COMPUTER NETWORK RESEARCH

ADVANCED RESEARCH PROJECTS AGENCY SEMIANNUAL TECHNICAL REPORT

June 30, 1971

Principal Investigator: Leonard Kleinrock

Co-Principal Investigators: Gerald Estrin
Michel Melkanoff
Richard R. Muntz

Computer Science Department
School of Engineering and Applied Science
University of California, Los Angeles

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) School of Engineering and Applied Science Computer Science Department 90024 405 Hilgard, University of California, Los Angeles	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b GROUP

3 REPORT TITLE

Computer Network Research

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

5 AUTHOR(S) (First name, middle initial, last name)

Leonard Kleinrock

6 REPORT DATE June 30, 1971	7a. TOTAL NO OF PAGES 130	7b. NO OF REFS 29
8a CONTRACT OR GRANT NO. b. PROJECT NO c. d	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10 DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY

13 ABSTRACT

ARPA Semiannual Technical Report, August 15, 1970, to June 30, 1971. (The length of this reporting period is due to an adjustment made to bring the Semiannual Technical Reports and Quarterly Management Reports into synchronization.)

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.C. Government.

Sponsored by

ADVANCED RESEARCH PROJECTS AGENCY

SEMIANNUAL TECHNICAL REPORT

June 30, 1971

COMPUTER NETWORK RESEARCH

ARPA Contract DAHC-15-69-C-0285

ARPA Order No. 1380

Principal Investigator: Leonard Kleinrock

Co-Principal Investigators: Gerald Estrin
Michel Melkanoff
Richard R. Muntz

Computer Science Department
School of Engineering and Applied Science
University of California, Los Angeles

UNCLASSIFIED

ADVANCED RESEARCH PROJECTS AGENCY

SEMIANNUAL TECHNICAL REPORT

June 30, 1971

Project Computer Network Research ARPA Order Number 1380

Contract Number DAHC-15-69-C-0285

Effective Date of Contract 4/1/69 Contract Expiration Date 3/31/72

Amount of Contract $1,723,932*

Contractor: School of Engineering and Applied Science
Computer Science Department
University of California
Los Angeles, California 90024

Project Scientist and Principal Investigator Dr. Leonard Kleinrock

Phone (213) 825-2543

---

*Following is a breakdown of amendments to the contract, all of which have been funded:

Period Covered	Amount	Annex
4/1/69-10/31/69	$229,300	A
11/1/69-10/31/70	$344,413	A
4/1/69-10/31/70	$ 69,396	A
12/1/69-6/30/70	230,000	D
7/1/70-6/30/71	300,000	D
11/1/70-1/31/71	60,000	A
2/1/71-3/31/72	490,823	A

TABLE OF CONTENTS

ADVANCED RESEARCH PROJECTS AGENCY

SEMIANNUAL TECHNICAL REPORT

June 30, 1971

1. INTRODUCTION

The goal of this project is to provide an environment for high quality research activities in information processing. One major area of research is mathematical modeling and analysis of computer systems. Another principle area of research, which is closely coupled with the first, is measurement of computer systems. We have been particularly active in the analysis and measurement of time-shared systems and the ARPA computer communication network. Our third major area of responsibility has been in the specification of software protocol for use in the network.

This report details our progress from the time of our last Semiannual Technical Report of August 15, 1970, through June 30, 1971.* References [1] to [15] include work accomplished prior to the current reporting period.

In Section 2 of this report, we survey our progress in the modeling and analysis of computer systems. In Section 2.1 we discuss the work on time-shared systems, and in Section 2.2 the work on computer-communication networks. A major effort has been our work on network measurement, and this is discussed in Section 3. Progress in the development of our time-sharing system and the network software development is described in Section 4. Section 5 concludes with some general comments about our progress.

---

*The length of this reporting period is due to an adjustment made to bring the Semiannual Technical Reports and Quarterly Management Reports into synchronization.

2. ANALYTIC MODELING AND MEASUREMENT OF COMPUTER SYSTEMS

Our research in computer systems modeling and measurement has been mainly in the areas of time-shared scheduling algorithms and computer-communication networks.

2.1. Time-Shared Systems Analysis

During this period we have made considerable progress in our research in the analysis of time-shared systems. In the previous Semiannual Technical Report we reported on efforts by Kleinrock and Muntz to analyze a very general class of scheduling algorithms. This class of scheduling algorithms includes as special cases most of the algorithms previously studied in the literature and also many additional algorithms. For example, it includes the case of a multilevel queueing system in which round-robin is used at some intermediate level queue (our SEX time-sharing system has such a scheduling algorithm). The method of solution required analyzing the round-robin system with bulk arrivals, and this was accomplished by Kleinrock, Muntz and Roderick (Ref. [16]). This paper forms Appendix A of this report. The application of these results to multilevel queueing systems is detailed in Ref. [17], which is included as Appendix B.

During this past year, we have also made progress in the area of bounds and approximations in the analysis of queuing systems. Recently, Kleinrock, Muntz and Hsu have succeeded in finding tight upper and lower bounds on the mean response time for a given amount of service time required in an M/G/1 queueing system with an arbitrary scheduling algorithm which does not make use of a priori knowledge of a customer's service time requirement. Given the arrival rate of customers and the service time distribution, bounds are determined such that the response function for any scheduling algorithm

must at all points lie between the upper and lower bounds. This is an important result since it places non-trivial limits on what can be accomplished by varying the scheduling algorithm for a system. Also in this work, several necessary conditions were found for feasible response curves in addition to the upper and lower bounds. The results of this work are to be reported at the 1971 IFIPS Congress, Ljubljana, Yugoslavia, August 1971 (Ref. [18]). This paper is included as Appendix C. We are continuing our efforts in this area with the aim of further characterizing feasible response functions.

In the last Semiannual Technical Report it was mentioned that a major unsolved problem in computer systems analysis is the consideration of multiple resources. Fouad Tobagi, a graduate student, is working in this area under the direction of Kleinrock. He is investigating the application of some recent results from the literature on approximation techniques to the problem of analyzing networks of queues. The results to date appear promising in providing a computationally efficient means of analyzing queueing networks in which there is a limiting resource or bottleneck.

During this reporting period, we have begun an effort in the area of measurement of time-shared computer systems. Johnny Wong, a graduate student working with Muntz, has begun a measurement project on the SEX time-sharing system. Based on measurements of process execution time requirements, swapping time and page requirements, a new scheduling algorithm was designed and is currently being implemented. Measurement of the paging behavior of processes has strongly suggested the possibility of increasing system efficiency by allowing processes to communicate page requirements to the operating system through system calls. A preliminary set of system calls has been implemented and is currently being evaluated. This appears to be virginal area for study.

2.2. Computer-Communication Nets

We have continued a strong effort in the area of computer-communications networks analysis and optimization. In particular, major areas of research are: routing, nodal blocking behavior in networks, and optimal assignment of channel capacity.

Gary Fultz, a graduate student working with Kleinrock, has used mathematical analysis and simulation to study adaptive routing techniques in store-and-forward computer networks. Using average message delay as a measure of network performance, a number of routing algorithms have been evaluated. The ability of the algorithms to adapt to communication line failures has been determined by simulation. A portion of this effort is described in Ref. [19] (Appendix D). Ref. [20] is nearly completed and will in addition report on analysis of the effects of multipacket messages and nodal storage requirements.

An important and difficult problem in store-and-forward networks is that of nodal blocking. When a node's buffer storage is filled, it becomes blocked and cannot receive new messages. This puts an increased load on this node's neighbors in the network and thus, nodal blocking is a transient effect which can propagate in time and space. Kleinrock and graduate student Jack Zeigler have studied this problem and their results are reported in Ref. [21] and Ref. [22]. Ref [21] is included as Appendix E.

Graduate students Mario Gerla and Luigi Fratta have worked with Kleinrock on computationally efficient techniques for determining the optimal assignment of channel capacity in a computer-communication network. The objective is either to minimize delay with the total cost held fixed or to minimize total cost with the delay held fixed. Under the conditions of

negligible nodal blocking and fixed network topology and routing, an optimization algorithm has been devised which is significantly more efficient than conventional techniques for constrained optimization problems. A report of this work is now in preparation (Ref. [23]). Further work in this area will include the network topology and routing as variables in the optimization.

At the IFIP Congress 71, Professor Wesley Chu will present a paper dealing with the selection of an optimal message block size for computer communications (Ref. [24], Appendix F). In this research he analyzes the relationships among acknowledgment time, channel transmission rate, channel error characteristics, average message length and optimal block size. Currently, Chu is completing a study of demultiplexing buffer requirements using a simulation model (Ref. [25]).

Professor Cantor has investigated the design of non-blocking switching networks with a minimum number of switches. The results of this study are included as Appendix G (Ref. [26]).

A paper surveying various aspects of the optimization of computer-communication networks was presented by Kleinrock at the 1971 IEEE National Convention (Ref. [27], Appendix H).

3. NETWORK MEASUREMENTS

The network measurement activity has involved a variety of tasks, including the further development of the measurement tools, "shakedown" tests on the network performance, measurement of actual user traffic, and the use of measurements to improve analytic models of the network behavior (Ref. [28]). Each of these areas is discussed in some detail in the following paragraphs. Gerald Cole has been the principal participant in this effort.

3.1. Extensions to the Measurement Capabilities

The control of network experiments and the collection of measurement data were originally developed to operate in a stand-alone (batch operating system) environment, but have been modified to also function under the SEX time-sharing system. This change allows one to conduct data gathering experiments along with the regular interactive usage of the system, and it provides the basis for further on-line data gathering and reduction usage. However, most of the experiments run during the reporting period utilized the earlier system due to the large computation overhead of the artificial traffic generator. This overhead is particularly large since the generator was modified to produce pseudo-random message lengths and interarrival times in addition to the earlier fixed parameter capabilities, but the random generation capabilities proved to be essential for many of the experiments which were conducted.

3.2. Analytical Efforts Related to Measurements

Some of the more significant results of the measurement efforts to date have involved the creation or improvement of analytic models of the network behavior based on insights gained from experimental measurement data. The modeling and measurement efforts were found to be quite complementary and resulted in an iterative procedure of model building and evaluation, with feedback from each test resulting in a more acceptable model. Models were developed in this manner relating to priority handling of messages, optimal packet sizes, and the separation of packets due to interference traffic. Several significant improvements were made in the models based on observed discrepancies between the observed and originally predicted behavior and resulted in good agreement for the refined models.

3.3. Network Experiments

In addition to the measurements related to analytic models as described above, several experiments were run to measure network usage and to attempt to predict the network performance. The first of these tests involved the measurement of the traffic between SRI and the University of Utah in December of 1970. Data were taken during several hours of the SRI usage of the PDP-10 at Utah, and these data were correlated with the known formats and activities involved in the transactions. In this manner, we were able to gain information on the user behavior, and at the same time, verify the operation and utility of the measurement routines.

One of the primary concerns in the analysis of the SRI-Utah traffic measurement was the matter of how many such users the network could simultaneously support. A rather crude estimate was made based on Scherr's* model of user think time and "processing" needs, and resulted in a range of 50 to 170 users depending on the file transmission requirements of each user. These interference tests were extended by use of artificial traffic and produced saturation levels which were consistent with the values as predicted by Kleinrock's results,** and led to further investigations of cyclic queueing phenomena associated with RFNM driven traffic on a given set of links. The through-put for such a condition was also investigated as a function of the number of links. This latter test resulted in an interesting demonstration of several of the measurement techniques in resolving a discrepancy between the expected and measured saturation through-put.

---

* Scherr, A.L., "An Analysis of Time-Shared Computer Systems," The MIT Press, 1967.

** Kleinrock, L., "Certain Analytic Results for Time-Shared Processors," Proc. IFIP Congress 1968, Edinburg, Scotland, pp. D119-D125, August 5-10, 1968.

3.4. Coordination with BBN

Several peculiar effects were encountered during the network experiments which were eventually found to be "bugs" in the network itself. Several of these effects were resolved in the IMP system that was released in mid-November, 1970, but others were subsequently encountered, particularly in regard to the handling and measurement of moderately high traffic loads. This latter problem became more visible after we requested that BBN change the round-trip delay recording resolution from 0.1 to 0.8 msec. to avoid a register overflow problem. Subsequent tests showed that the IMP would "crash" at certain traffic levels, and BBN was then able to isolate and eliminate the problem. The network control center personnel were quite helpful in these efforts and also cooperated in the execution of some of the subsequent tests, e.g., by changing selected IMP parameter values during a test.

3.5. Measurement Plans

Measurement plans for the near future include the monitoring of network usage as the new protocol becomes operational and conducting a set of "before and after" tests to determine the effect of the BBN changes in the flow control and routing algorithms which will soon be implemented. Other experiments will also be run to further evaluate and improve some of the analytic models, and to check out new data reduction programs as they become available.

4. NETWORK AND SYSTEMS SOFTWARE

This section covers work done by the SPADE Group which has been under the leadership of Steve Crocker and Jon Postel. The effort has been divided equally between maintaining and extending the SEX time-sharing system and development of network software.

4.1. Network Progress

Three major meetings of the network Working Group (NWG) were held. Steve Crocker of UCLA was Chairman and major organizer of these meetings. In conjunction with the Fall Joint Computer Conference, the NWG met in Houston in November. Discussion there centered on the problems of console interaction between systems, particularly the incompatibility of line-oriented local echo devices with character-oriented remote echo systems. It was agreed that this incompatibility would prevent some users with line-oriented consoles from using some character-oriented systems, but this could be tolerated. The NWG held a February meeting at the University of Illinois. Network protocols were the topic of discussion. The primary concern was over some needed refinements to the HOST-HOST or level 2 protocol. A special committee chaired by Steve Crocker was set up to resolve these issues and its report (NWG/RFC #107) is an official modification to the protocol. This report calls for new command formats, new commands (Reset, Reset reply), replacement of marking with a fixed size header, and the introduction of byte sizes. Also discussed at the Illinois meeting were the use of sockets and the initial connection protocol. A third NWG meeting was held at Atlantic City in conjunction with the Spring Joint Computer Conference in May. Prime topics of discussion at this meeting were several 3rd level protocols, e.g., Telnet-Logger, File Transfer, and initial connection procedures. Committees were established to deal with each of these topics. The Telnet-Logger issues were resolved at the May meeting and initial connection protocol was established in early June.

Implementation of a Network Control Program (NCP) and Telnet and Logger programs which follow the official specifications are now completed and oper-

ational on our time-sharing system.

The SPADE Group has provided support for the measurement experiments conducted by Gerald Cole. The measurement programs can now be run under the SEX system in parallel with other network and local use of the system.

4.2. System Development

The SEX time-sharing system has grown to support more of the users of the Sigma-7. We have acquired two IMLAC PDS-1 display terminals and four model 33 teletypes. The operating system has been changed in several ways. Some of the changes are corrections to defects in the system as acquired from LRL. The file system and resident operating system were made more independent so that a system crash no longer causes the file system to be destroyed. A garbage collection process now reclaims lost file space and forces file system consistency. An interprocess communication facility called Events was added. A batch processing facility has been implemented to provide service for non-interactive users and a tape input-output supervisor has been implemented. An operator's control program has been implemented which allows the selective starting and stopping of system level programs, e.g., NCP, printer process. The system is currently scheduled for standard user service 20 hours per week. Reliability has been improved to the point that system crashes now occur on the average of once per week.

5. CONCLUSIONS AND SELF-EVALUATION

Our efforts in the mathematical modeling and measurement of computer systems has been very profitable during this period. In the computer networks area particularly, our two-pronged attack with analysis and measurement

has yielded significant results. We have begun a measurement effort in connection with time-shared systems and plan to accelerate this effort in cooperation with our analytic work in this area. Progress on the development of network protocol has been substantial. Developmental work on the SEX time-sharing system has not diminished as much as had been expected, but it has been a necessary and worthwhile investment. We plan to continue shifting more of our efforts toward the modeling and measurement areas.

Our efforts have established UCLA as a leader in the field of modeling and analysis of computer systems and an important member of the ARPA Network community.

REFERENCES

1. Kleinrock, L. "Time-Sharing Systems: Analytical Methods," Proc. of the Symposium on Critical Factors in Data Management/1968, UCLA, March 20-22, 1968, Prentice-Hall, pp. 3-32, 1969.

2. Martin, D., and G. Estrin. "Path Length Computations on Graph Models of Computations," Transactions of the IEEE, Vol. C-18, pp. 530-536, June 1969.

3. Kleinrock, L. "Models for Computer Networks," Proc. of the IEEE International Conference on Communications, Boulder, Colo., pp. 21-16 to 21-29, June 9-11, 1969.

4. Kleinrock, L. "On Swap Time in Time-Shared Systems," Proc. of the IEEE Computer Group Conference, Minneapolis, Minn., pp. 37-41, June 17-19, 1969.

5. Coffman, E.G., Jr., and R. R. Muntz. "Model of Pure Time-Sharing Disciplines for Resource Allocation," Proc. of the 24th National Conference of ACM, August 1969.

6. Chu, W. W. "A Study of Asynchronous Time Division Multiplexing for Time-Sharing Computer Systems," 1970 Proc. of the Fall Joint Computer Conference, Las Vegas, Nev., pp. 669-678, November 1969.

7. Kleinrock, L. "Comparison of Solution Methods for Computer Network Models," Proc. of the Computer and Communications Conference, Rome, N.Y., Oct. 2, 1969.

8. Muntz, R. R., and R. Uzgalis. "Dynamic Storage Allocation for Binary Search Trees in a Two-Level Memory," Proc. of the Fourth Annual Princeton Conference on Information Sciences and Systems, Princeton, N.J., March 26-27, 1970.

9. Kleinrock, L. "A Continuum of Time-Sharing Scheduling Algorithms," Proc. of the 1970 Spring Joint Computer Conference, Atlantic City, N.J., pp. 453-458, May 1970.

10. Kleinrock, L. "Analytic and Simulation Methods in Computer Network Design," Proc. of the 1970 Spring Joint Computer Conference, Atlantic City, N.J., pp. 569-579, May 1970.

11. Carr, C. S., S. D. Crocker, and V. G. Cerf. "HOST-HOST Communication Protocol in the ARPA Network," Proc. of the 1970 Spring Joint Computer Conference, Atlantic City, N.J., pp. 589-597, May 1970.

12. Chu, W. W. "Selection of Optimal Transmission Rate for Statistical Multiplexors," Proc. of the 1970 IEEE International Conference on Communications, San Francisco, Calif., pp. 28-22 to 28-25, June 8-10, 1970.

13. Kleinrock, L. "Swap Time Considerations in Time-Shared Systems," IEEE Transactions on Computers, pp. 534-540, June 1970.

14. Kleinrock, L., and R. R. Muntz. "Multilevel Processor-Sharing Queueing Models for Time-Shared Systems," Proc. of the Sixth International Teletraffic Congress, Munich, Germany, pp. 341/1-341/8, August 1970.

15. Chu, W. W. "Buffer Behavior for Batch Poisson Arrivals and Single Constant Output," IEEE Trans. on Communication Technology, October 1970.

16. Kleinrock, L., R. R. Muntz, and E. Rodemich. "The Processor-Sharing Queueing Model for Time-Shared Systems with Bulk Arrivals," to be published in Networks, Vol. I, No. 1.

17. Kleinrock, L., and R. R. Muntz. "Processor-Sharing Queueing Models of Mixed Scheduling Disciplines for Time-Shared Systems," to be published in J. Assoc. Computing Machinery.

18. Kleinrock, L., R. R. Muntz, and J. Hsu. "Tight Bounds on the Average Response Time for Time-Shared Computer Systems," to be presented at the 1971 International Federation for Information Processing, Ljubljana, Yugoslavia, August 23-28, 1971.

19. Fultz, G., and L. Kleinrock. "Adaptive Routing Techniques for Store-and-Forward Computer-Communication Networks," 1971 International Conference on Communications, Montreal, Canada, June 14-16, 1971.

20. Fultz, G. "Adaptive Routing Techniques for Store-and-Forward Message Switching Computer-Communication Networks," Ph.D. dissertation, School of Engineering and Applied Science, Computer Science Department, University of California, Los Angeles, 1971.

21. Zeigler, J., and L. Kleinrock. "Nodal Blocking in Large Networks," 1971 International Conference on Communications, Montreal, Canada, June 14-16, 1971.

22. Zeigler, J. "Nodal Blocking in Large Networks," Ph.D. dissertation, School of Engineering and Applied Science, Computer Science Department, University of California, Los Angeles, 1971.

23. Fratta, L., M. Gerla, and L. Kleinrock. "The Slow Deviation Method: A New Approach to the Analysis and Synthesis of Store-and-Forward Communication Networks," to be published in Networks.

24. Chu, W. W. "Optimal Fixed Message Block Size for Computer Communications," to be presented at the 1971 International Federation for Information Processing, Ljubljana, Yugoslavia, August 23-28, 1971.

25. Chu, W. W. "Demultiplexing Considerations for Statistical Multiplexors," to be presented at the ACM Second Symposium on the Problems in the Optimization of Data Communication Systems," Palo Alto, Calif., Oct. 18-20, 1971.

26. Cantor, D. G. "On Non-Blocking Switching Networks," to be published in Networks, Vol. 1, No. 1.

27. Kleinrock, L. "Delay in Communication and Computer Networks," IEEE 71 International Convention and Exposition, New York, March 22-25, 1971.

28. Cole, G. D. "Computer Network Measurements: Techniques and Experiments," Ph.D. dissertation, School of Engineering and Applied Science, Computer Science Department, University of California, Los Angeles, 1971

29. Hsu, J. "Analysis of a Continuum of Processor-Sharing Models for Time-Shared Computer Systems," Ph.D. dissertation, School of Engineering and Applied Science, Computer Science Department, University of California, Los Angeles, 1971.

Presentations

1. Kleinrock, L. "Survey of Analytical Results of Time-Shared Computer Systems," North Carolina State University, Raleigh, Oct. 27, 1970.

2. Kleinrock, L. "Time-Shared Systems Analysis," University of Illinois, Urbana, Oct. 28, 1970.

3. Kleinrock, L. "Time-Shared Systems Analysis," Operations Research Society of American Conference, Detroit, Mich., Oct. 20, 1970.

4. Kleinrock, L. "Recent Results in Time-Shared Systems Analysis," Applied Physics and Information Science Department, University of California, San Diego, Nov. 16, 1970.

5. Kleinrock, L. "Survey of Analytical Methods in Queueing Networks," New York University, Nov. 30, 1970.

6. Kleinrock, L. "Graphs and Networks in the Real World," International Conference on Circuit Theory, Atlanta, Ga., Dec. 16, 1970.

7. Kleinrock, L. Attended ARPA Contractors Meeting, San Diego, Calif., Feb. 9-11, 1971.

8. Kleinrock, L. "Time-Sharing System Modeling," ACM SIGTIME Group, Systems Development Corporation, Santa Monica, Calif., March 8, 1971.

9. Muntz, R. "Bounds on the Response Characteristics of Queueing Systems," IBM Yorktown Heights, N.Y., April 2, 1971.

10. Kleinrock, L., and R. R. Muntz. "Processor-Sharing Queueing Models of Mixed Scheduling Disciplines for Time-Shared Systems," 39th Meeting of Operations Society of America, Dallas, Tex., May 5-7, 1971.

APPENDICES

APPENDIX A

THE PROCESSOR-SHARING QUEUEING MODEL FOR TIME-SHARED SYSTEMS WITH BULK ARRIVALS

by L. Kleinrock, R. R. Muntz and E. Rodemich

THE PROCESSOR-SHARING QUEUEING MODEL FOR TIME-SHARED SYSTEMS WITH BULK ARRIVALS*

by L. Kleinrock, R. R. Muntz

Computer Science Department
University of California, Los Angeles, California

and

E. Rodemich

Jet Propulsion Laboratory
Pasadena, California

ABSTRACT

We consider a model which is applicable to time-multiplexed systems, such as multiplexed communication channels and time-shared computing facilities. In this (processor-sharing) queueing model, all jobs currently in the system share equally the processing capability of the server. In this paper, we investigate the processor-sharing model for the case of bulk arrivals. The mean response time of the system as a function of required service time is derived. An example is given to show the effect of bulk arrivals versus single arrivals for a constant utilization.

---

* This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285). This paper also presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored by the National Aeronautics and Space Administration.

THE PROCESSOR-SHARING QUEUEING MODEL FOR TIME-SHARED SYSTEMS WITH BULK ARRIVALS*

by L. Kleinrock, R. R. Muntz
Computer Science Department
University of California, Los Angeles, California

and

E. Rodemich
Jet Propulsion Laboratory
Pasadena, California

I. INTRODUCTION

In a time-sharing system, the computing facilities are time-multiplexed among the currently active jobs according to some scheduling discipline. A major goal of the scheduling discipline is to provide short response times to small requests for service. This creates an effective environment for interaction between a user at a console and the computing facility since most interactive requests are for relatively small amounts of service. The user should expect longer delays if his request is for a significant amount of service. Analytic studies of these systems are aimed at determining the effect of various scheduling disciplines on response time.

In computer networks, we are often faced with a configuration in which many time-shared computer systems are interconnected over a communication network. It is important to understand the behavior of these time-shared nodes so that one can evaluate the performance of these networks.

The application of queueing models to time-shared computer systems has been an active area of research since 1964 [1]. A survey of this area

---

* This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285). This paper also presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored

is available in reference [2]. In this paper, we generalize some previous models by permitting bulk arrivals to the system.

In the usual round-robin scheduling discipline, a newly arriving job must join the end of a queue for the server. When it reaches the front of the queue, it is allocated a quantum of time on the server. If the job completes before the quantum expires, it leaves the system. Otherwise, it must rejoin the end of the queue to wait for its next quantum. In this paper, the quantum size is allowed to shrink to zero so that we have the "processor-sharing" discipline. In effect, each job receives $1/n^{th}$ of the processing capability of the processor when there are n jobs demanding service. The model is illustrated in Fig. 1. This processor-sharing discipline was first introduced in 1967 [3] and analyzed for the case of single Poisson arrivals. In this paper, we consider the bulk arrival case where customers may arrive in groups. The arrival instants are, as usual, assumed to be Poisson.

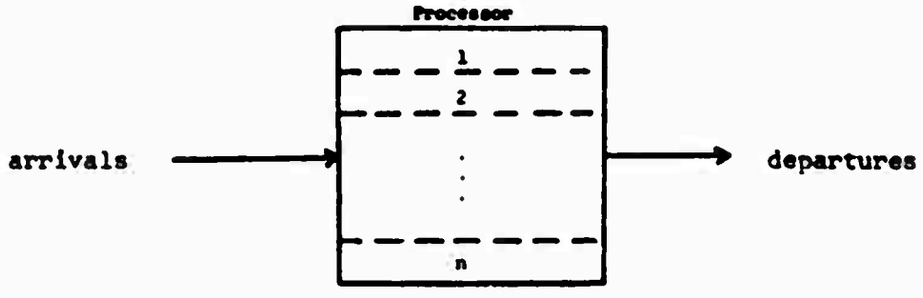


Figure 1. The processor-sharing model

Aside from the case where bulk arrivals may be the actual arrival mechanism for the system, the bulk arrival case presented here arises naturally in solving more general models. Consider the case where the server becomes unavailable for periods of time with distribution F(d) and mean $\bar{d}$. When the server once again becomes available, he finds that a number of new customers have joined the system and so it appears to him that a bulk arrival has taken place. Assume also that the interval between the end of one of these "down" times and the start of the next is exponentially distributed with mean $\bar{u}$.

The queueing time for the job can be thought of as the sum of two intervals: the time between the arrival instant and the first time the server becomes available, and from this time until the job leaves the system. The mean length of the first interval is easily calculated to be $\frac{\overline{d^2}}{2(\bar{d} + \bar{u})}$. To find the mean length of the second interval we use the following approach. We telescope the time axis to reduce all of the server "down" intervals to zero length but keep the number of arrivals the same in each interval. On the new time axis, the set of arrivals leaving a down time appears as a bulk

arrival. The generating function for the bulk size is easily calculable. Now solve for the mean waiting time, $\bar{W}_V$ for a job in this "virtual" time frame. The virtual waiting time and the actual waiting time, $\bar{W}_A$, are related by $\bar{W}_A = \frac{\bar{W}_V}{1 - \mu\bar{d}}$. The mean queueing time for a job is just the sum of the means of the two intervals described.

Other situations in which servicing of a class of jobs is interrupted can be approached in this manner. For example, priority queueing models with arbitrary service disciplines. The approach has already been applied to the analysis of feedback queueing models for time-shared systems [4].

2. MODEL

Formally, the parameters of the model are:

1) the interarrival times are exponentially distributed with the mean interarrival time λ, i.e.,

$$P[\text{interarrival time} \leq t] = 1 - e^{-\lambda t} \qquad t \geq 0$$

2) the distribution of required service time in the CPU, B(t) is assumed to be of the form $B(t) = \begin{cases} 1 - q(t)e^{-\beta t} & 0 \leq t < t_1 \\ 0 & t \geq t_1 \end{cases}$

where q(t) is a polynomial in t of degree n. This of course, includes the exponential service time distribution when q(t) = 1.

3) the distribution of a, the bulk size is arbitrary with mean $\bar{a}$.

The generating function of a is $G(z) = \sum_{i=0}^{\infty} z^i P[a = i]$.

4) the queueing discipline is processor-sharing, i.e., the limit of the round-robin discipline as the quantum size approaches zero.

We want to solve for the mean queueing time to service a job requiring τ seconds of service. This is denoted by $T(\tau)$.

3. AN INTEGRAL EQUATION DESCRIBING AVERAGE RESPONSE TIME

We approach this problem by first considering a discrete time system with quantum size $q > 0$. We assume that arrivals and departures take place only at times that are integral multiples of q. For small q any continuous distribution can be approximated. By letting q approach 0 equations for continuous time systems can be found.

Let $n(iq)$ = the mean number of jobs in the system with iq seconds of attained service when a tagged job arrives.

σ_i = the probability that a job which has received iq seconds of service will require more than $(i + 1)q$ seconds of service.

$\bar{a}$ = the mean bulk size of arrivals.

b = the mean number of arrivals with a tagged job.

Since we intend to let q approach 0, the position of the tagged job with respect to the jobs that arrive in the same group is not important. We will assume for convenience that the tagged job is the last job in the group.

The mean time until the tagged job has received its first quantum of service is given by

$$T_1 = \sum_{j=0}^{\infty} n(jq)q + bq + q$$

In general, the mean time between the $(i-1)^{st}$ and i^{th} quantum of service to the tagged job is given by

$$T_i = \sum_{j=0}^{\infty} (n(jq)\sigma_j\sigma_{j+1} \cdots \sigma_{j+i-2}q)$$

$$+ \sum_{j=1}^{i-1} (\lambda\bar{a}\, T_j \sigma_0\sigma_1 \cdots \sigma_{i-j-2}q)$$

$$+ q + b[\sigma_0\sigma_1 \cdots \sigma_{i-1}]q \qquad (1)$$

The first term represents the time required by those jobs which were initially in the system and will still be there after the tagged job has received $i-1$ quanta of service. The second term is the contribution due to jobs that have arrived since the tagged job entered the system. The third term is due to the tagged job itself. The last term results from those jobs which arrived with the tagged job and require more than $i-1$ quanta of service.

Dividing both sides of Eq. (1) by q we get

$$\frac{T_i}{q} = \sum_{j=0}^{\infty} n(jq)\sigma_j\sigma_{j+1} \cdots \sigma_{j+i-2}$$

$$+ \sum_{j=1}^{i-1} \lambda\bar{a}\, T_j \sigma_0\sigma_1 \cdots \sigma_{i-j-2}$$

$$+ 1 + b\sigma_0\sigma_1 \cdots \sigma_{i-1} \qquad (2)$$

Let $iq = \tau$ and $jq = x$. Then as $q \to 0$:

$$\frac{T_i}{q} \to T'(\tau) \equiv \frac{dT(\tau)}{dt}$$

$$\sigma_j \sigma_{j+1} \cdots \sigma_{j+i-2} \to \frac{1 - B(x + \tau)}{1 - B(x)}$$

$$n(jq) \to n(x)$$

$$\sigma_0 \sigma_1 \cdots \sigma_{i-j-2} \to 1 - B(\tau - x)$$

$$\sigma_0 \sigma_1 \cdots \sigma_{i-1} \to 1 - B(\tau)$$

Therefore as $q \to 0$ Eq. (2) becomes

$$T'(x) = \int_0^\infty n(x) \frac{1 - B(x + \tau)}{1 - B(x)} dx$$

$$+ \lambda \bar{a} \int_0^\tau T'(x)[1 - B(\tau - x)]dx$$

$$+ 1 + b[1 - B(\tau)]$$

From Kleinrock and Coffman [6] we also have that

$$n(x) = \lambda \bar{a}[1 - B(x)]T'(x)$$

Substituting for $n(x)$ we have

$$T'(\tau) = \lambda \bar{a} \int_0^\infty T'(x)[1 - B(x + \tau)]dx$$

$$+ \lambda \bar{a} \int_0^\tau T'(x)[1 - B(\tau - x)]dx$$

$$+ 1 + b[1 - B(\tau)] \qquad (3)$$

In terms of the generating function G(z) for a, we have that $\bar{a} = G'(1)$. The value of b can also be expressed in terms of G(z). Consider that the tagged job is selected at random from the arrivals to the queueing system. Then the probability that the job is selected from a bulk size of n jobs is given by $\frac{nP[a = n]}{\bar{a}}$ [5].

Therefore $b + 1 = \sum_{n=0}^{\infty} n \frac{nP[a = n]}{\bar{a}} = \frac{E[a^2]}{\bar{a}} = \frac{G''(1) + G'(1)}{G'(1)}$ or $b = \frac{G''(1)}{G'(1)}$.

It remains only to solve the integral equation, Eq. (3).

4. SOLUTION OF THE INTEGRAL EQUATION

In this section we solve the integral equation (3) for the average response time $T(\tau)$. Recall that we have restricted the service time distribution B(t) such that

$$\text{With } 1 - B(t) = \begin{cases} e^{-\beta t} q(t) & o < t < t_1 \\ o & t \geq t_1 \end{cases}$$

where q(t) is a polynomial of degree n.

Then Eq. (3) becomes

$$T'(\tau) = \lambda\bar{a} \int_o^{t_1 - \tau} T'(x) q(x + \tau) e^{-\mu(x+\tau)} dx$$

$$+ \lambda\bar{a} \int_o^{\tau} T'(x) q(\tau - x) e^{-\mu(\tau - x)} dx$$

$$+ bq(\tau) e^{-\mu\tau} + 1 \qquad (4)$$

After multiplying Eq. (4) by $e^{\mu\tau}$, it may be rewritten as

$$e^{\mu\tau}T'(\tau) = \lambda\bar{a}\int_0^{t_1-\tau} e^{-\mu x}q(\tau + x)T'(x)dx$$

$$+ \lambda\bar{a}\int_0^{\tau} e^{\mu x}q(\tau - x)T'(x)dx$$

$$+ bq(\tau) + e^{\mu\tau} \qquad (5)$$

Let D denote the differential operator $d/d\tau$. Differentiating Eq. (5) $n + 1$ times, we get

$$D^{n+1}[e^{\mu\tau}T'(\tau)] = \lambda\bar{a}\sum_{k=0}^{n} q^{(k)}(0)D^{n-k}[e^{\mu\tau}T'(\tau)]$$

$$- \lambda\bar{a}\sum_{k=0}^{n} e^{-\mu t_1}q^{(k)}(t_1)D^{n-k}[e^{\mu\tau}T'(t_1 - \tau)] + \mu^{n+1}e^{\mu\tau}$$

where $q^{(k)}$ is the k^{th} derivative of q. Now, multiplying by $e^{-\mu\tau}$, the result can be put in the form

$$O_o(D)T'(\tau) + O_1(D)T'(t_1 - \tau) = \mu^{n+1} \qquad (6)$$

where $O_o(D)$ and $O_1(D)$ are linear differential operators with constant coefficients, given by the following formulas:

$$O_o(\xi) = (\xi + \mu)^{n+1} - \lambda\bar{a}\sum_{k=0}^{n} q^{(k)}(0)(\xi + \mu)^{n-k}$$

$$O_1(\xi) = \lambda\bar{a}\sum_{k=0}^{n} e^{-\mu t_1}q^{(k)}(t_1)(\xi + \mu)^{n-k}$$

Replacing τ by $t_1 - \tau$ in Eq. (6), we get

$$Q_1(-D)T'(\tau) + Q_0(-D)T'(t_1 - \tau) = \mu^{n+1}$$

Apply $-Q_1(D)$ to this equation, $Q_0(-D)$ to (2), and add. Then we have

$$Q_2(D)T'(\tau) = [Q_0(0) - Q_1(0)]\mu^{n+1} \tag{7}$$

where

$$Q_2(\xi) = Q_0(\xi)Q_0(-\xi) - Q_1(\xi)Q_1(-\xi)$$

Since $Q_2(\xi)$ is unchanged when ξ is replaced by $-\xi$, its roots occur in pairs $(a_m, -a_m)$, $m = 1, \ldots, n+1$, and

$$Q_2(\xi) = (-1)^{n+1} \prod_{m=1}^{n+1} (\xi^2 - a_m^2)$$

For general $q(\tau)$, these roots are distinct and non-zero. Then the general solution of Eq. (7) is

$$T'(\tau) = \alpha_0 + \sum_{m=1}^{n+1} (A_m e^{-a_m \tau} + B_m e^{a_m \tau}) \tag{8}$$

where the constant α_0 is given by

$$\alpha_0 = [Q_0(0) - Q_1(0)]\mu^{n+1}/Q_2(0)$$

which can be reduced to

$$\alpha_0 = \frac{1}{1 - \lambda\bar{a}\bar{t}}$$

where $\bar{t}$ is the mean of $B(t)$.

A formula for $T'(t_1 - \tau)$ follows from Eq. (8) by replacing τ by $t_1 - \tau$. Using these expressions in Eq. (6), and equating the coefficient of each exponential to zero, we get the conditions

$$B_m Q_o(\alpha_m) + A_m e^{-\alpha_m t_1} Q_1(\alpha_m) = 0, \quad m = 1, \ldots, n+1$$

which we can satisfy by putting

$$A_m = C_m Q_o(\alpha_m)$$

$$B_m = -C_m e^{-\alpha_m t_1} Q_1(\alpha_m), \qquad m = 1, \ldots, n+1$$

Then

$$T'(\tau) = \frac{1}{1 - \lambda \bar{a} \bar{t}} + \sum_{m=1}^{n+1} C_m [Q_o(\alpha_m) e^{-\alpha_m \tau} - Q_1(\alpha_m) e^{-\alpha_m (t_1 - \tau)}] \tag{9}$$

This expression for $T'(\tau)$ must be put in the original integral Eq. (4) to determine the coefficients $C_1, \ldots, C_{n+1}$. Collecting the coefficients of the various exponentials which arise, we get the following system of equations:

$$\lambda \bar{a} \sum_{m=1}^{n+1} C_m [Q_o(\alpha_m) + e^{-\alpha_m t_1} Q_1(\alpha_m)] \left| \frac{1}{(u - \alpha_m)^j} - \frac{1}{(u + \alpha_m)^j} \right|$$

$$= \begin{cases} b, & j = 1 \\ 0, & j = 2, \ldots, n+1 \end{cases} \tag{10}$$

In expressing the solution of this system by determinants, only the determinant of the coefficients, and the cofactors of the coefficients with $j = 1$ are needed. These can be given explicitly:

$$\Delta = \det \left[\frac{1}{(\mu - \alpha_m)^j} - \frac{1}{(\mu + \alpha_m)^j} \right]_{m,j=1,\ldots,n+1}$$

$$= \frac{2^{\frac{1}{2}(n+2)(n+1)} (-1)^{\frac{1}{2}n(n+1)} \alpha_1 \cdots \alpha_{n+1} \mu^{\frac{1}{2}n(n+1)} \prod_{j>k} (\alpha_j^2 - \alpha_k^2)}{\prod_{k=1}^{n+1} (\mu^2 - \alpha_k^2)^{n+1}}$$

and a typical cofactor is

$$\Delta_{11} = \det \left[\frac{1}{(\mu - \alpha_m)^j} - \frac{1}{(\mu + \alpha_m)^j} \right]_{m,j=2,\ldots,n+1}$$

$$= \frac{2^{\frac{1}{2}(n+2)(n+1)} (-1)^{\frac{1}{2}n(n+1)} \alpha_2 \cdots \alpha_{n+1} \mu^{\frac{1}{2}n(n+1)} \prod_{j>k\geq 2} (\alpha_j^2 - \alpha_k^2)}{2 \prod_{k=2}^{n+1} (\mu^2 - \alpha_k^2)^{n+1}}$$

We have

$$\lambda \bar{a} C_1 [O_o(\alpha_1) + e^{-\alpha_1 t_1} O_1(\alpha_1)] = b\Delta_{11}/\Delta$$

$$= \frac{b(\mu^2 - \alpha_1^2)^{n+1}}{2\alpha_1 \prod_{k=2}^{n+1} (\alpha_k^2 - \alpha_1^2)}$$

$$= -b(\mu^2 - \alpha_1^2)^{n+1} / Q_2'(\alpha_1)$$

and the other coefficients have similar formulas. Using these in Eq. (9),

$$T'(\tau) = \frac{1}{1 - \lambda\bar{a}\bar{t}} - \frac{b}{\lambda\bar{a}} \sum_{m=1}^{n+1} \frac{(\mu^2 - \alpha_m^2)^{n+1}}{Q_2'(\alpha_m)} \cdot \frac{Q_o(\alpha_m)e^{-\alpha_m \tau} - Q_1(\alpha_m)e^{-\alpha_m (t_1 - \tau)}}{Q_o(\alpha_m) + e^{-\alpha_m t_1} Q_1(\alpha_m)} \qquad (11)$$

This last equation is the solution to Eq. (3) which we are seeking. It is interesting to observe that for the non-bulk arrival case (i.e., $P[a = 1] = 1$), Eq. (11) reduces to

$$T'(\tau) = \frac{1}{1 - \rho}$$

where

$$\rho = \lambda\bar{a}/\mu$$

This is the well-known result for single Poisson arrival to a round robin processor sharing system with arbitrary service time distribution [7].

5. AN EXAMPLE

Let $q(t) = 1$ and $t_1 = \infty$ so that the service times are exponentially distributed. Then

$$Q_o(\xi) = \xi + \mu - \lambda\bar{a}$$

$$Q_1(\xi) = 0$$

$$Q_2(\xi) = \mu^2 - 2\mu\lambda\bar{a} + \lambda^2\bar{a^2} - \xi^2$$

The roots of $Q_2(\xi)$ are $\pm (\mu - \lambda\bar{a}) = \pm \alpha_1$

Therefore

$$T'(\tau) = \frac{1}{1 - \lambda\bar{a}/\mu} - \frac{b}{\lambda\bar{a}} \left[\frac{(\mu^2 - \alpha_1^2)}{-2\alpha_1} \quad \frac{(\alpha_1 + \mu - \lambda\bar{a})e^{-\alpha_1 \tau}}{\alpha_1 + \mu - \lambda\bar{a}} \right]$$

$$T(\tau) = \frac{\tau}{1 - \lambda\bar{a}/\mu} - \frac{b}{\lambda\bar{a}} \left[\frac{(\mu^2 - \alpha_1^2)(\alpha_1 + \mu - \lambda\bar{a})\left[e^{-\alpha_1 \tau} - 1\right]}{(-2\alpha_1)(-\alpha_1)(\alpha_1 + \mu - \lambda\bar{a})} \right]$$

or

$$T(\tau) = \frac{\tau}{1 - \lambda\bar{a}/\mu} + \frac{b}{\lambda\bar{a}} \left[\frac{(\mu^2 - (\mu - \lambda\bar{a})^2)(1 - e^{-(\mu - \lambda\bar{a})\tau})}{2(\mu - \lambda\bar{a})^2} \right]$$

Figure 2 shows this average response function for the case $\lambda = 0.75$, $\mu = 1.0$, $\bar{a} = 0.385$, $b = 0.746$. The parameters in this example were chosen to correspond with an example from Reference [4]. Also shown is the solution for the non-bulk arrival case with the same service time distribution and the same mean arrival rate.

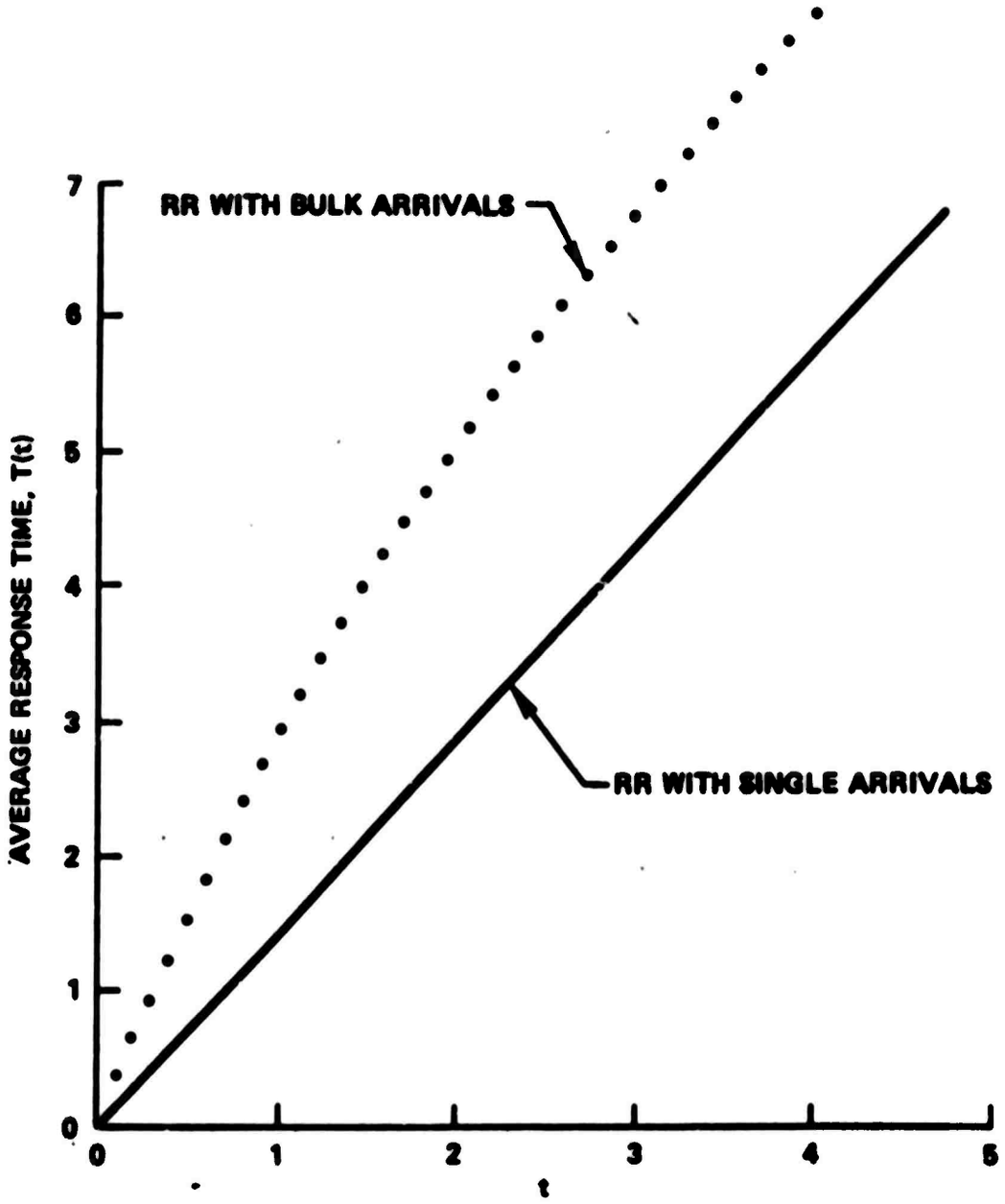


Figure 2. A comparison of Average Response Times for the Bulk Arrival and Single Arrival Cases with Exponential Service Times;
Single Arrivals: $\lambda = 0.288$, $\mu = 1.0$, $\rho = 0.288$
Bulk Arrivals: $\lambda = 0.75$, $\mu = 1.0$, $\rho = 0.788$, $\bar{s} = 0.388$, $b = .746$

6. CONCLUSION

The processor-sharing queueing model with bulk arrivals has been studied. An integral equation, Eq. (3), describing the mean queueing time for a job requiring τ seconds of service was developed which is valid for arbitrary service time distributions. This integral equation was solved in Section 4 for a class of service time distributions which includes the exponential distribution. The application of the bulk arrival model to queueing systems with periods of server unavailability were indicated in Section 1.

REFERENCES

1. Kleinrock, L., "Analysis of a Time-Shared Processor," Naval Research Logistics Quarterly, Vol. II, No. 1, pp. 59-73, March 1964.

2. McKinney, J. M., "A Survey of Analytical Time-Sharing Models," Computing Surveys, Vol. 1, No. 2, June 1969, pp. 105-116.

3. Kleinrock, L., "Time-Shared Systems: A Theoretical Treatment," JACM, Vol. 14, No. 2, April 1967, pp. 242-261.

4. Kleinrock, L., and R. R. Muntz, "Multilevel Processor-Sharing Queueing Models for Time-Shared Systems," Proc. of the Sixth International Teletraffic Congress, Munich, Germany, pp. 341/1-341/8, August 1970.

5. Avi-Itzhak, B., W. L. Maxwell, and L. W. Miller, "Queueing with Alternating Priorities," Operations Research 13, No. 2, 1965.

6. Kleinrock, L., and E. G. Coffman, "Distribution of Attained Service in Time-Shared Systems," J. of Computers and Systems Science, Vol. 3, October 1967, pp. 287-298.

7. Sakata, M., S. Noguchi, and J. Oizumi, "Analysis of a Processor-Shared Queueing Model for Time-Sharing Systems," Proc., 2nd Hawaii International Conf. on System Sciences, January 1969, pp. 625-628.

APPENDIX B

PROCESSOR-SHARING QUEUEING MODELS OF MIXED SCHEDULING DISCIPLINES FOR TIME-SHARED SYSTEMS

by L. Kleinrock and R. R. Muntz

PROCESSOR-SHARING QUEUEING MODELS OF MIXED SCHEDULING DISCIPLINES FOR TIME-SHARED SYSTEMS*

by L. Kleinrock and R. R. Muntz

Computer Science Department
University of California, Los Angeles, California

ABSTRACT

Scheduling algorithms for time-shared computing facilities are considered in terms of a queueing theory model. The extremely useful limit of "processor-sharing" is adopted, wherein the quantum of service shrinks to zero; this approach greatly simplifies the problem. A class of algorithms is studied for which the scheduling discipline may change for a given job as a function of the amount of service received by that job. These multilevel disciplines form a natural extension to many of the disciplines previously considered.

The average response time for jobs conditioned on their service requirement is solved for in this paper. Explicit solutions are given for the system M/G/1 in which levels may be first-come-first-served (FCFS), feedback (FB) or round-robin (RR) in any order. The service time distribution is restricted to be a polynomial times an exponential for the case of RR.

Examples are described for which the average response time is plotted. These examples display the great versatility of the results and demonstrate the flexibility available for the intelligent design of discriminatory treatment among jobs (in favor of short jobs and against long jobs) in time-shared computer systems.

---

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285).

PROCESSOR-SHARING QUEUEING MODELS OF MIXED SCHEDULING DISCIPLINES FOR TIME-SHARED SYSTEMS*

by L. Kleinrock and R. R. Muntz

Computer Science Department
University of California, Los Angeles, California

1. INTRODUCTION

Queueing models have been used successfully in the analysis of time-shared computer systems since the appearance of the first applied paper in this field in 1964 [1]. A recent survey of this work is given by McKinney [2]. One of the first papers to consider the effect of feedback in queueing systems was due to Takacs [3].

One of the goals in a time-shared computer system is to provide rapid response to those tasks which are inter-active and which place frequent, but small, demands on the system. As a result, the system scheduling algorithm must identify those demands which are small, and provide them with preferential treatment over larger demands. Since we assume that the scheduler has no explicit knowledge of job processing times, this identification is accomplished implicitly by "testing" jobs. That is, jobs are rapidly provided small amounts of processing and, if they are short, they will depart rather quickly; otherwise, they will linger while other, newer jobs are provided this rapid service, etc., thus providing good response to small demands.

Generally, an arrival enters the time-shared system and competes for the attention of a single processing unit. This arrival is forced to wait

---

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285).

in a system of queues until he is permitted a quantum of service time; when this quantum expires, he is then required to join the system of queues to await his second quantum, etc. The precise model for the system is developed in Section 2. In 1967 the notion of allowing the quantum to shrink to zero was studied [4] and was referred to as "processor-sharing", in 1966 Schrage [18] also studied the zero-quantum limit. As the name implies, this zero-quantum limit provides a share or portion of the processing unit to many customers simultaneously; in the case of round-robin (RR) scheduling [4], all customers in the system simultaneously share (equally or on a priority basis) the processor, whereas in the feedback (FB) scheduling [5] only that set of customers with the smallest attained service share the processor. We use the term processor-sharing here since it is the processing unit itself (the central processing unit of the computer) which is being shared among the set of the customers; the phrase "time-sharing" will be reserved to imply that customers are waiting sequentially for their turn to use the entire processor for a finite quantum. In studying the literature one finds that the obtained results appear in a rather complex form and this complexity arises from the fact that the quantum is typically assumed to be finite as opposed to infinitesimal. When one allows the quantum to shrink to zero, giving rise to a processor-sharing system, then the difficulty in analysis as well as in the form of results disappears in large part; one is thus encouraged to consider the processor-sharing case. Clearly, this limit of infinitesimal quantum* is an ideal and can seldom be reached in practice due to overhead considerations; nevertheless, its extreme simplicity in analysis and results brings us to the studies reported in this paper.

The two processor-sharing systems studied in the past are the RR and

---

*This limiting case is not unlike the diffusion approximation for queues (see, for example, Gaver [15]).

the FB [4,5]. Typically, the quantity solved for is $T(t)$, the expected response time conditioned on the customer's service time t; response time is the elapsed time from when a customer enters the system until he leaves completely serviced. This measure is especially important since it exposes the preferential treatment given to short jobs at the expense of the long jobs. Clearly, this discrimination is purposeful since, as stated above, it is the desire in time-shared systems that small requests should be allowed to pass through the system quickly. In 1969 the distribution for the response time in the RR system was found [6]. In this paper we consider the case of mixed scheduling algorithms whereby customers are treated according to the RR algorithms, the FB algorithm, or first come first served (FCFS) algorithm, depending upon how much total service time they have already received. Thus, as a customer proceeds through the system obtaining service at various rates he is treated according to different disciplines; the policy which is applied among customers in different levels is that of the FB system as explained further in Section 2. Thus, natural generalization of the previously studied processor-sharing systems allows one to create a large number of new and interesting disciplines whose solutions we present.

A more restricted study of this sort was reported by the authors in [16]. Here we make use of the additional results from [11] to generalize our analysis.

2. THE MODEL

The model we choose to use in representing the scheduling algorithms is drawn from queueing theory. This corresponds to the many previous models studied [1,2,4,5,6,7,8,18], all of which may be thought of in terms of the

structure shown in Fig. 2.1. In this figure we indicate that new requests enter the system in queues upon arrival. Whenever the computer's central processing unit (CPU) becomes free, some customer is allowed into the service facility for an amount of time referred to as a quantum. If, during this quantum, the total accumulated service for a customer equals his required service time, then he departs the system; if not, at the end of his quantum, he cycles back to the system of queues and waits until he is next chosen for additional service. The system of queues may order the customers according to a variety of different criteria in order to select the next customer to receive a quantum. In this paper, we assume that the only measure used in evaluating this criterion is the amount of attained service (total service so far received).

In order to specify the scheduling algoirthm in terms of this model, it is required that we identify the following:

a. The customer interarrival time distribution. We assume this to be exponential, i.e.,

$$P[\text{interarrival time} \leq t] = 1 - e^{-\lambda t} \qquad t \geq 0 \qquad (2.1)$$

where λ is the average arrival rate of customers.

b. The distribution of required service time in the CPU. This we assume to be arbitrary (but independent of the interarrival times). We thus assume

$$P[\text{service time} \leq x] = B(x) \qquad (2.2)$$

Also assume $1/\mu$ = average service time.

c. The quantum size. Here we assume a processor-shared model in which customers receive an equal but vanishingly small amount of service each time they are allowed into service. For more discussion of such systems, see [4,6,7,18].

d. The system of queues. We consider here a generalization and consolidation of many systems studied in the past. In particular, we define a set of attained service times $\{a_i\}$ such that

$$0 = a_0 < a_1 < a_2 < \ldots < a_N < a_{N+1} = \infty \tag{2.3}$$

The discipline followed for a job when it has attained service, τ, in the interval

$$a_{i-1} \leq \tau < a_i \qquad i = 1,2,\ldots,N+1 \tag{2.4}$$

will be denoted as D_i. We consider D_i for any given level i to be either: FIRST COME FIRST SERVED (FCFS); PROCESSOR SHARED-FB$_\infty$ (FB); or ROUND-ROBIN PROCESSOR SHARED (RR). The FCFS system needs no explanation, the FB system gives service next to that customer who so far has least attained service; if there is a tie (among K customers, say) for this position, then all K members in the tie get served simultaneously (each attaining useful service at a rate of $1/K$ sec/sec), this being the nature of processor sharing systems. The RR processor sharing system shares the service facility among all customers present (say J customers) giving attained service to each at a rate of $1/J$ sec/sec. Moreover, between intervals, the jobs are treated as a set of FB disciplines (i.e., service proceeds in the i^{th} level only if all levels $j < i$ are empty). See Fig. 2.2. For example, for $N = 0$, we have the usual single-level case of either FCFS, RR, or FB. for $N = 1$, we could have any of nine disciplines (FCFS followed by FCFS, ..., RR followed by RR); note that FB followed by FB is just a single FB system (due to overall FB policy between levels).

As an illustrative example, consider the $N = 2$ case shown in Fig. 2.3. Any new arrivals begin to share the processor in a RR fashion with all other customers who so far have less than 2 seconds of attained service.

Customers in the range of $2 \leq \tau < 6$ may get served only if no customers present have had less than 2 seconds of service; in such a case, that customer (or customers) with the least attained service will proceed to occupy the service in an FB fashion until they either leave, or reach $\tau = 6$, or some new customer arrives (in which case the overall FB rule provides that the RR policy at level 1 preempts). If all customers have $\tau \geq 6$, then the "oldest" customer will be served to completion unless interrupted by a new arrival. The history of some customers in this example system is shown in Fig. 2.4. We denote customer n by C_n. Note that the slope of attained service varies as the number of customers simultaneously being serviced changes. We see that C_2 requires 5 seconds of service and spends 14 seconds in system (i.e., response time of 14 seconds).

So much for the system specification. We may summarize by saying that we have an M/G/1 queueing system* model with processor sharing and with a generalized multilevel scheduling structure.

The quantity we wish to solve for is

$$T(t) = E[\text{response time for a customer requiring a total of } t \text{ seconds of attained service}] \qquad (2.5)$$

We further make the following definitions:

$$T_i(t) = E[\text{time spent in interval } i\ [a_{i-1}, a_i) \text{ for customers requiring a total of } t \text{ seconds of attained service}] \qquad (2.6)$$

---

*M/G/1 denotes the single-server queueing system with Poisson arrivals and arbitrary service time distribution; similarly M/M/1 denotes the more special case of exponential service time distribution. One might also think of our processor-sharing system as an infinite server model with constant overall service rate.

We note that

$$T_i(t) = T_i(t') \qquad \text{for } t,t' \geq a_i \tag{2.7}$$

Furthermore, we have, for $a_{k-1} \leq t < a_k$, that

$$T(t) = \sum_{i=1}^{k} T_i(t) \tag{2.8}$$

Also we find it convenient to define the following quantities with respect to $B(t)$:

$$\bar{t}_{cx} = \int_0^x t\,dB(t) + x \int_x^\infty dB(t) \tag{2.9}$$

$$\bar{t^2}_{cx} = \int_0^x t^2\,dB(t) + x^2 \int_x^\infty dB(t) \tag{2.10}$$

$$\rho_{cx} = \lambda \bar{t}_{cx} \tag{2.11}$$

$$W_x = \frac{\lambda \bar{t^2}_{cx}}{2(1 - \rho_{cx})} \tag{2.12}$$

Note that W_x represents the expected work found by a new arrival in the system M/G/1 where the service times are truncated at x.

3. RESULTS FOR MULTILEVEL QUEUEING SYSTEMS

We wish to find an expression for $T(t)$, the mean system time (i.e., average response time) for jobs with service time t such that $a_{i-1} \leq t < a_i$, i.e., jobs which reach the i^{th} level queue and there leave the system. To accomplish this it is convenient to isolate the i^{th} level to some extent. We make use of the following two facts.

1. By the assumption of preemptive priority of lower level queues (i.e., FB discipline between levels) it is clear that jobs in levels higher than the i^{th} level can be ignored. This follows since these jobs cannot interfere with the servicing of the lower levels.

2. We are interested in jobs that will reach the i^{th} level queue and then depart from the system before passing to the $(i + 1)^{st}$ level. The system time of such a job can be thought of as occurring in two parts. The first portion is the time from the job's arrival to the queueing system until the group at the i^{th} level is serviced for the first time after this job has reached the i^{th} level. The second portion starts with the end of the first portion and ends when the job leaves the system. It is easy to see that both the first and second portions of the job's system time are unaffected by the service disciplines used in levels 1 through $i - 1$. Therefore, we can assume any convenient disciplines. In fact, all these levels can be lumped into one equivalent level which services jobs with attained service between 0 and a_{i-1} seconds using any service discipline.

From (1) and (2) above it follows that we can solve for $T(t)$ for jobs that leave the system from the i^{th} level by considering a two-level system. The lower level services jobs with attained service between 0 and a_{i-1} whereas the second level services jobs with attained service between a_{i-1}

and a_i. Jobs that would have passed to the $i+1^{st}$ level after receiving a_i seconds of service in the original system are now assumed to leave the system at that point. In other words the service times are truncated at a_i.

3.1 i^{th} Level Discipline is FB

Consider the two-level system with the second level corresponding to the i^{th} level of the original system. Since we are free to choose the discipline used in the lower level, we can assume that the FB discipline is used in this level as well. Clearly the two-level system behaves like a single level FB system with service times truncated at a_i. The solution for such a system is known [5,9]:

$$T(t) = \frac{t}{1 - \rho_{<t}} + \frac{\lambda \overline{t^2}_{<t}}{2(1 - \rho_{<t})^2} \qquad (3.1)$$

3.2 i^{th} Level Discipline is FCFS

Consider again the two-level system with breakpoints at a_{i-1} and a_i. Regardless of the discipline in the lower level, a tagged job entering the system will be delayed by the sum of a) the work currently in both levels ($= W_{a_i}$) plus, b) any new arrivals to the lower level queue during the interval (average $T(t)$) this job is in the system. These new arrivals form a Poisson process with parameter λ and their contribution to the delay is a random variable whose first and second moments are $\bar{t}_{<a_{i-1}}$ and $\overline{t^2}_{<a_{i-1}}$ respectively. Thus we have

$$T(t) = W_{a_i} + \lambda \bar{t}_{a_{i-1}} T(t) + t$$

and so

$$T(t) = \frac{W_{a_i} + t}{1 - \rho_{<a_{i-1}}} \tag{3.2}$$

where W_{a_i} is given by Eq. (2.12). It is also possible to use these methods for solving last-come-first-served and random order of service at any level.

3.3 i^{th} Level Discipline is RR

In this case, our results are limited in the i^{th} interval to service time distributions in which

$$B(x) = 1 - p(x)e^{-\beta x} \qquad a_{i-1} \leq x < a_i \tag{3.3}$$

$$p(x) = p_o + p_1 x + \ldots + p_n x^n \tag{3.4}$$

The service time distribution $F(x)$, for this i^{th} interval is then

$$F(x) = \begin{cases} \dfrac{B(a_{i-1} + x) - B(a_{i-1})}{1 - B(a_{i-1})} = 1 - q(x)e^{-\beta x} & o \leq x < a_i - a_{i-1} \\ 1 & x \geq a_i - a_{i-1} \end{cases} \tag{3.3a}$$

where

$$q(x) = \frac{e^{-\beta a_{i-1}} p(a_{i-1} + x)}{1 - B(a_{i-1})} = q_o + q_1 x + \ldots + q_n x^n \tag{3.4a}$$

Thus we permit in this interval, service time distributions of the form: 1 minus a polynomial of degree n times an exponential. The analysis of this system appears in [11]; we make use of these results below. Nevertheless, we develop our analysis as far as possible for the case of general $B(x)$ before specializing to the class given by Eqs. (3.3) (3.4).

We start by considering the two-level system with breakpoints at a_{i-1} and a_i. Consider the busy periods of the lower level. During each such busy period there may be a number of jobs that pass to the higher level. We choose to consider these arrivals to the higher level as occurring at the end of the lower level busy period so that there is a bulk arrival to the higher level at this time. We also choose to temporarily delete these lower level busy periods from the time axis. In effect we create a virtual time axis telescoped to delete the lower level busy periods. Since the time from the end of one lower level busy period to the start of the next is exponentially distributed (Poisson arrivals!), the arrivals to the higher level appear in virtual time to be bulk arrivals at instants generated from a Poisson process with parameter λ.

Consider a tagged job that required $t = a_{i-1} + \tau$ seconds of service ($0 < \tau \leq a_i - a_{i-1}$). Let α_1 be the mean real time the job spends in the system until its arrival (at the end of the lower level busy period) at the higher level queue. Let $\alpha_2(\tau)$ be the mean virtual time the job spends in the higher level queue.

α_1 can be calculated as follows. The initial delay is equal to the mean work the job finds in the lower level on arrival plus the a_{i-1} seconds of work that it contributed to the lower level. Therefore

$$\alpha_1 = W_{a_{i-1}} + \lambda \bar{t}_{a_{i-1}} \alpha_1 + a_{i-1}$$

and so

$$\alpha_1 = \frac{1}{1 - \rho_{<a_{i-1}}} \left(W_{a_{i-1}} + a_{i-1} \right) \tag{3.5}$$

If $\alpha_2(\tau)$ is the mean virtual time the job spends in the higher level, we can easily convert this to the mean real time spent in this level. In the virtual time interval $\alpha_2(\tau)$ there are an average of $\lambda\alpha_2(\tau)$ lower level busy periods that have been ignored. Each of these has a mean length of $\frac{\bar{t}_{<a_{i-1}}}{1-\rho_{<a_{i-1}}}$. Therefore, the mean real time the job spends in the higher level is given by

$$\alpha_2(\tau) + \lambda\alpha_2(\tau)\frac{t_{<a_{i-1}}}{1-\rho_{<a_{i-1}}} = \frac{\alpha_2(\tau)}{1-\rho_{<a_{i-1}}} \qquad (3.6)$$

Combining these results we see that a job requiring $t = a_{i-1} + \tau$ seconds of service has mean system time given by

$$T(a_{i-1}+\tau) = \frac{1}{1-\rho_{<a_{i-1}}}\left\{W_{a_{i-1}} + a_{i-1} + \alpha_2(\tau)\right\} \qquad (3.7)$$

The only unknown quantity in this equation is $\alpha_2(\tau)$. To solve for $\alpha_2(\tau)$ we must, in general, consider an M/G/1 system with bulk arrival and RR processor sharing. The only exception is the case of RR at the first level which has only single arrivals. Since the higher level queues can be ignored, the solution in this exceptional case is the same as for a round-robin single level system with service times truncated at a_1. Therefore, from [8] we have for the first level

$$T(t) = \frac{t}{1-\rho_{<a_1}} \qquad 0 \le t < a_1 \qquad (3.8)$$

Let us now consider the bulk arrival RR system in isolation in order to solve for the virtual time spent in the higher level queue, $\alpha_2(\tau)$. The service time distribution for this bulk arrival system is

$$F(x) = \begin{cases} \dfrac{B(a_{i-1} + x) - B(a_{i-1})}{1 - B(a_{i-1})} & 0 \leq x < a_i - a_{i-1} \\ 1 & x \geq a_i - a_{i-1} \end{cases}$$

Note that the utilization factor for this bulk system is

$$\rho = \lambda \bar{a} \bar{x} / \mu_i \tag{3.9}$$

where $\bar{a}$ is the mean number of arrivals in a bulk and $\frac{1}{\mu_i}$ is the mean of the distribution $F(x)$. Let us begin by solving for $\bar{a}$. This we do for the general case a_{i-1}, a_i. $\bar{a}$ is just the mean number of jobs that arrive during a low level busy period and require more than a_{i-1} seconds of service. Therefore $\bar{a}$ must satisfy the equation

$$\bar{a} = \lambda \bar{t}_{<a_{i-1}} \bar{a} + [1 - B(a_{i-1})]1 \tag{3.10}$$

In this equation $\lambda t_{<a_{i-1}}$ is the mean number of jobs that arrive during the service time of the first job in the busy period. Since each of these jobs in effect generates a busy period, there are an average of $\lambda \bar{t}_{<a_{i-1}} \bar{a}$ arrivals to the upper level queue due to these jobs. The second term is just the average number of times that the first job in the busy period will require more than a_{i-1} seconds of service.

Clearly then

$$\bar{a} = \frac{1 - B(a_{i-1})}{1 - \rho_{<a_{i-1}}} \tag{3.11}$$

In [11], an integral equation is derived which defines $\alpha_2(\tau)$ for the RR bulk arrival system; we repeat that equation below

$$a_2'(\tau) = \lambda\bar{a}\int_0^{\infty} a_2'(x)[1 - F(x + \tau)]dx$$

$$+ \lambda\bar{a}\int_0^{\tau} a_2'(x)[1 - F(\tau - x)]dx$$

$$+ 1 + b[1 - F(\tau)] \tag{3.12}$$

where $a_2'(\tau) = da_2(\tau)/d\tau$, and b = mean number of arrivals with the tagged job. The solution to this integral equation for the restricted service time distributions as given in Eqs. (3.3a) and (3.4a) is also given in [11], and for our problem takes the form

$$a_2(\tau) = \frac{\tau}{1 - \lambda\bar{a}\frac{1}{\mu_1}} - \frac{b}{\lambda\bar{a}}\sum_{m=1}^{n+1}\frac{\left(\beta^2 - \gamma_m^2\right)^{n+1}}{Q_2'(\gamma_m)}$$

$$\cdot \frac{Q_0(\gamma_m)[1 - e^{-\gamma_m\tau}] - Q_1(\gamma_m)e^{-\gamma_m x_1}[e^{\gamma_m\tau} - 1]}{[Q_0(\gamma_m) + e^{-\gamma_m x_1}Q_1(\gamma_m)]\gamma_m} \tag{3.13}$$

where

$$x_1 = a_i - a_{i-1} \tag{3.14}$$

$$Q_0(x) = (x + \beta)^{n+1} - \lambda\bar{a}\sum_{k=0}^{n} q^{(k)}(0)(x + \beta)^{n-k} \tag{3.15}$$

$$Q_1(x) = \lambda\bar{a}\sum_{k=0}^{n} e^{-\beta x_1}q^{(k)}(x_1)(x + \lambda)^{n-k} \tag{3.16}$$

$$Q_2(x) = Q_0(x)Q_0(-x) - Q_1(x)Q_1(-x) \tag{3.17}$$

and where Eq. (3.17) has roots (occurring in pairs) $x = -\gamma_m, \gamma_m$ for $m = 1, 2, \ldots, n+1$ and the notation $f^{(k)}(\gamma)$ denotes the k^{th} derivative of f with respect to its argument evaluated at a value γ.

In the solution for $\alpha_2(\tau)$ given in Eq. (3.13), we are required to compute b(= mean number of arrivals with a tagged job). This we do by first deriving an expression for

$$G(z) \equiv \sum_{k=0}^{\infty} P[\text{bulk size} = k] z^k \tag{3.18}$$

which is the probability generating function (z-transform) for the bulk size. Either by direct arguments based upon busy periods or by use of the method of collective marks [12], we readily arrive at

$$G(z) = [1 - B(a_{i-1})]z \sum_{j=0}^{\infty} \frac{(\lambda a_{i-1})^j}{j!} e^{-\lambda a_{i-1}} [G(z)]^j$$

$$+ B(a_{i-1}) \left[\int_0^{a_{i-1}} \sum_{j=0}^{\infty} \frac{(\lambda t)^j}{j!} e^{-\lambda t} [G(z)]^j \frac{dB(t)}{B(a_{i-1})} \right] \tag{3.19}$$

In Eq. (3.19) the first term is conditioned on the assumption that the customer who preempts service from those at level i reaches the i^{th} level; the second term assumes that he does not reach level i. Eq. (3.19) reduces to

$$G(z) = [1 - B(a_{i-1})]z e^{-\lambda a_{i-1}[1-G(z)]} + \int_0^{a_{i-1}} e^{-\lambda t[1-G(z)]} dB(t) \tag{3.20}$$

For arbitrary $B(x)$, we cannot reduce this last expression any further. Nevertheless, we can obtain moments from it. In particular, from the definition of $\bar{a}$, we obtain

$$\bar{a} = G'(z)\Big|_{z=1} = \frac{1 - B(a_{i-1})}{1 - \lambda \bar{t}_{<a_{i-1}}}$$

which is exactly as obtained by more direct arguments in Eq. (3.11). However, we are seeking b. For this we must calculate

$$G''(z)\Big|_{z=1} = \frac{(\bar{a})^2}{1 - \rho_{<a_{i-1}}} \left[2\lambda a_{i-1}(1 - \rho_{<a_{i-1}}) + \lambda^2 \overline{t^2}_{<a_{i-1}} \right] \qquad (3.21)$$

Now since the mean group size (1 + b) of a tagged customer's group is related to the bulk size distribution as the mean spread is related to the inter-event distribution (namely the mean spread equals the second moment of the inter-event interval divided by the first moment) [13], we have

$$1 + b = \frac{\text{second moment of bulk size}}{\text{first moment of bulk size}} \qquad (3.22)$$

or

$$b = \frac{G''(z)}{G'(z)}\Big|_{z=1} \qquad (3.23)$$

From Eq. (3.20) we get

$$b = \frac{\bar{a}}{1 - \rho_{<a_{i-1}}} \left[2\lambda a_{i-1}(1 - \rho_{<a_{i-1}}) + \lambda^2 \overline{t^2}_{<a_{i-1}} \right] \qquad (3.24)$$

Having solved for $\alpha_2(\tau)$ we may now substitute back into Eq. (3.7) which solves the case when the i^{th} level discipline is RR and service time is of the form given in Eqs. (3.3a) and (3.4a). (Note that for $i = 1$, the solution given in Eq. (3.8) is good for any $B(x)$.)

It is instructive to display the solution for $T(t)$ explicitly in a special case for our i^{th} level RR discipline. We choose the multilevel system with M/M/1 and solve for $T(a_{i-1} + \tau)$ after substituting $\alpha_2(\tau)$ into Eq. (3.7). Note for M/M/1 that $q(t) = q_o = 1$. Also, from Eqs. (3.14 - 3.17), and choosing $\beta = \mu$,

$$Q_o(x) = x + \mu - \lambda\bar{a} \tag{3.25}$$

$$Q_1(x) = \lambda\bar{a}e^{-\mu x_1} \tag{3.26}$$

$$Q_2(x) = \mu^2 - 2\mu\lambda\bar{a} + (\lambda\bar{a})^2(1 - e^{-2\mu x_1}) - x^2 \tag{3.27}$$

thus the roots of $Q_2(x)$ are

$$\pm\gamma_1 = \pm\sqrt{\mu^2 - 2\mu\lambda\bar{a} + (\lambda\bar{a})^2(1 - e^{-2\mu x_1})} \tag{3.28}$$

and

$$\frac{1}{\bar{\mu}_i} = \frac{1}{\mu}(1 - e^{-\mu x_1}) \tag{3.29}$$

thus from these and Eq. (3.13), we get

$$\alpha_2(\tau) = \frac{\tau}{1 - \lambda\bar{a}\frac{1}{\bar{\mu}_i}} + \frac{b(\mu^2 - \gamma_1^2)\left[(\gamma_1 + \mu - \lambda\bar{a})(1 - e^{-\gamma_1\tau}) - \lambda\bar{a}e^{-(\mu+\gamma_1)x_1}(e^{\gamma_1\tau} - 1)\right]}{2\lambda\bar{a}\gamma_1^2\left[\gamma_1 + \mu - \lambda\bar{a}(1 - e^{-(\mu+\gamma_1)x_1})\right]} \tag{3.30}$$

Also from Eqs. (2.9) and (2.10) we obtain

$$\bar{t}_{<x} = \frac{1}{\mu}(1 - e^{-\mu x}) \tag{3.31}$$

$$\overline{t^2}_{<x} = \frac{2}{\mu^2}(1 - e^{-\mu x} - \mu x e^{-\mu x}) \tag{3.32}$$

We may substitute these last two equations into Eqs. (3.11) and (3.24) to obtain a and b explicitly. Also, we note from Eqs. (2.12) and (3.32) that

$$W_{a_{i-1}} = \frac{\lambda(1 - e^{-\mu a_{i-1}} - \mu a_{i-1} e^{-\mu a_{i-1}})}{\mu^2[1 - \frac{\lambda}{\mu}(1 - e^{-\mu a_{i-1}})]} \tag{3.33}$$

Finally, we may substitute this expression for $W_{a_{i-1}}$ and Eq. (3.30) which gives $\alpha_2(\tau)$ into Eq. (3.7) which gives us the explicit form for $T(\tau)$.

4. EXAMPLES

In this section we demonstrate through examples the nature of the results we have obtained. Recall that we have given explicit solutions for our general model in the case M/G/1 with processor sharing where the allowed scheduling disciplines within a given level may be FCFS or FB; if the discipline is RR, it may be at level 1 and if it occurs at level $i > 1$, must be of the form given in Eqs. (3.3a) and (3.4a).

We begin with four examples from the system M/M/1. As mentioned in Section 2, we have nine disciplines for the case $N = 1$. This comes about since we allow any one of three disciplines at level 1 and any one of three disciplines at level 2. As we have shown, the behavior of the average conditional response time in any particular level is independent of the discipline in all other levels; thus we have nine disciplines. In Fig. 4.1

we show the behavior of each of the nine disciplines for the system $N = 1$. In this case we have assumed $\mu = 1$, $\lambda = 0.75$, and $a_1 = 2$. From Eq. (3.1) we see that the response time for the FB system is completely independent of the values a_i and therefore the curve shown in Fig. 4.1 for this response time is applicable to all of our M/M/1 cases. Note the inflection point in this curve and that the response time grows linearly as $t \to \infty$ (a phenomenon not observable from previously published figures but easily seen from Eq. (3.1)). As can be seen from its defining equation, the response time for FCFS is linear regardless of the level; the RR system at level 1 is also linear, but as we see from this figure and from Eq. (3.13) the RR at levels $i > 1$ is nonlinear. Thus one can determine the behavior of any of nine possible disciplines from Fig. 4.1. Adiri and Avi-Itzhak considered the case (FB, RR) [14].

Continuing with the case M/M/1, we show in Fig. 4.2 the case for $N = 3$ where D_1 = RR, D_2 = FB, D_3 = FCFS, and D_4 = RR. In this case we have chosen $a_i = i$ and $\mu = 1$, $\lambda = 0.75$. We also show in this figure the case FB over the entire range as a reference curve for comparison with this discipline. Note (in general for M/M/1) that the response time for any discipline in an given level must either coincide with FB curve or lie above it in the early part of the interval and below it in the latter part of the interval; this is true due to the conservation law [15].

The third example for M/M/1 is for the iterated structure D_i = FCFS. Once again we have chosen $\mu = 1$, $\lambda = 0.75$, and $a_i = i$. Also shown in this figure is a dashed line corresponding to the FB system over the entire range. Clearly, one may select any sequence of FB and FCFS with duplicates in adjacent intervals and the behavior for such systems can be found from Fig. 4.3. It is interesting to note in the general M/G/1 case with D_i = FCFS

that we have a solution for the FB system with finite quantum $q_i = a_i = a_{i-1}$ where preemption within a quantum is permitted!

Our fourth example is for an M/M/1 system with D_i = RR and is shown in Fig. 4.4. Here we use the explicit form for $T(\tau)$ derived from Eqs. (3.7), (3.30), and (3.33). We maintain the same value $\mu = 1$, $\lambda = 0.75$, $a_1 = 2$, $a_2 = 5$. $T(\tau)$ for this sytem is shown in Fig. 4.4.

We leave M/M/1 now and give two examples for M/G/1. For the first example we choose the system $M/E_2/1$ with $N = 1$. In this system we have

$$\frac{dB(x)}{dx} = (2\mu)^2 x e^{-2\mu x} \qquad x \geq 0 \tag{4.1}$$

We note that the mean service time here is again given by $1/\mu$; the second moment of this distribution is $3/2\mu^2$. We calculate

$$\bar{t}_{<a_1} = \frac{1}{\mu} - \frac{1}{\mu} e^{-2\mu a_1}[1 + 2\mu a_i + 2(\mu a_1)^2] \tag{4.2}$$

We choose the system $N = 1$ with D_1 = RR and D_2 = FCFS. For the cases $a_i = 1/2\mu$, $1/\mu$, $2/\mu$, $4/\mu$, and ∞ with $\mu = 1$ and $\lambda = 0.75$ we show in Fig. 4.5 the behavior of this system.

The last example we use is for the following service time distribution:

$$b_1(x) \equiv \frac{dB_1(x)}{dx} = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ e^{-2(x - \frac{1}{2})} & \frac{1}{2} \leq x \end{cases} \tag{4.3}$$

as shown in Fig. 4.6. In this case, $\bar{t}_{\leq \frac{1}{2}} = \frac{3}{8}$, $\overline{t^2}_{\leq \frac{1}{2}} = \frac{1}{6}$, $\bar{t} = \frac{5}{8}$, $\overline{t^2} = \frac{2}{3}$.

We choose the system D_1 = FCFS, D_2 = RR, and D_3 = FCFS with $a_1 = \frac{1}{2}$, $a_2 = \frac{3}{2}$, and $\lambda = 0.75$. The performance of this system is given in Fig. 4.7.

These examples demonstrate the broad behavior obtainable from our results as one varies the appropriate system parameters. In all cases the system discriminates in favor of the short jobs and against the longer jobs.

5. CONCLUSION

Our purpose has been to generalize results in the modelling and analysis of time-shared systems. The class of systems considered was the processor-sharing systems in which various disciplines were permitted at different levels of attained service. The principle results for M/G/1 are the following: (1) the performance for the FB discipline at any level is given by Eq. (3.1); (2) the performance for the FCFS discipline is linear with t within any level and is given by Eq. (3.2); (3) the performance for the RR discipline at the first level is well known [8] and is given by Eq. (3.8); (4) an integral equation for the average conditional response time for RR at any level (equivalent to bulk arrival RR) is given by Eq. (3.12) and remains unsolved in general; however, for the service distribution given in Eqs. (3.3a) and (3.4a), we have the general solution given in Eq. (3.13) as derived in [11]. We further note that the average conditional response time at level i is independent of the queueing discipline at all other levels.

Examples are given which display the behavior of some of the possible system configurations. From these, we note the great flexibility available in the multilevel systems. From the examples in Section 4, we see that considerable variation from previously studied algorithms is possible so long as the number of levels is less than a small integer (say 5); however, we see that as N increases, the behavior of the ML systems rapidly approaches

that of the pure FB system.

Examination of the envelope of the multitude of response functions available with the ML system has suggested that upper and lower bounds in system performance exist; this in fact has been established and is reported in [19].

REFERENCES

1. Kleinrock, L., "Analysis of a Time-Shared Processor," Naval Research Logistics Quarterly, Vol. II, No. 1, pp. 59-73, March 1964.

2. McKinney, J. M., "A Survey of Analytical Time-Sharing Models," Computing Surveys, Vol. 1, No. 2, June 1969, pp. 105-116.

3. Takács, L., "A Single-Server Queue with Feedback," Bell System Technical Journal, March 1963, pp. 505-519.

4. Kleinrock, L., "Time-Shared Systems: A Theoretical Treatment," JACM, Vol. 14, No. 2, April 1967, pp. 242-261.

5. Coffman, E. G., and L. Kleinrock, "Feedback Queueing Models for Time-Shared Systems," JACM, Vol. 15, No. 4, October 1968, pp. 549-576.

6. Coffman, E. G., Jr., R. R. Muntz, and H. Trotter, "Waiting Time Distributions for Processor-Sharing Systems," JACM, Vol. 17, No. 1, Jan. 1970, pp. 123-130.

7. Kleinrock, L., and E. G. Coffman, "Distribution of Attained Service in Time-Shared Systems," J. of Computers and Systems Science, Vol. 3, October 1967, pp. 287-298.

8. Sakata, M., S. Noguchi, and J. Oizumi, "Analysis of a Processor-Shared Queueing Model for Time-Sharing Systems," Proc., 2nd Hawaii International Conf. on System Sciences, Jan. 1969, pp. 625-628.

9. Schrage, L. E., "The Queue M/G/1 with Feedback to Lower Priority Queues," Management Science, Vol. 13, No. 7, 1967.

10. Conway, R. W., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley, 1967.

11. Kleinrock, L., R. R. Muntz, and E. Rodemich, "The Processor-Sharing Queueing Model for Time-Shared Systems with Bulk Arrivals," to be published in Networks Journal.

12. Cohen, J., The Single Server Queue, Wiley 1969.

13. Oliver, R. M., and W. S. Jewell, "The Distribution of Spread," Research Report 20, Operations Research Center, University of California, Berkeley, Calif., Jan. 25, 1962.

14. Adiri, I., and B. Avi-Itzhak, "Queueing Models for Time-Sharing Service Systems," Operations Research, Statistics and Economics Mimeograph Series No. 27, Technion, Israel.

15. Kleinrock, L., "A Conservation Law for a Wide Class of Queueing Disciplines," Naval Research Logistics Quarterly, Vol. 12, No. 2, June 1965, pp. 181-192.

16. Kleinrock, L., and R. R. Muntz, "Multilevel Processor-Sharing Queueing Models for Time-Shared Systems," Proc. of the Sixth International Teletraffic Congress, Munich, Germany, pp. 341/1-341/8, August 1970.

17. Gaver, D., "Diffusion Approximations and Models for Certain Congestion Problems," J. of Applied Probability, Vol. 5, 1968.

18. Schrage, L.E., "Some Queueing Models for a Time-Shared Facility," Ph.D. dissertation, Dept. of Industrial Engineering, Cornell University, 1966.

19. Kleinrock, L., R.R. Muntz and J. Hsu, "Tight Bounds on the Average Response Time for Processor-Sharing Models of Time-Shared Computer Systems," to be published in the Proc. of IFIPS Congress, 1971.

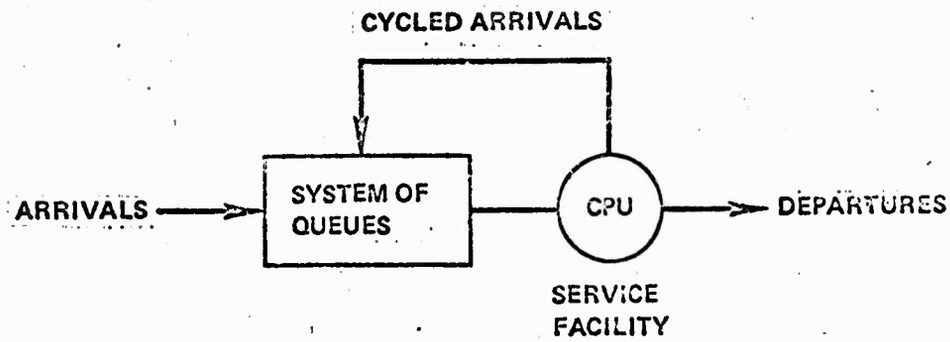


Figure 2.1. The Feedback Queueing Model

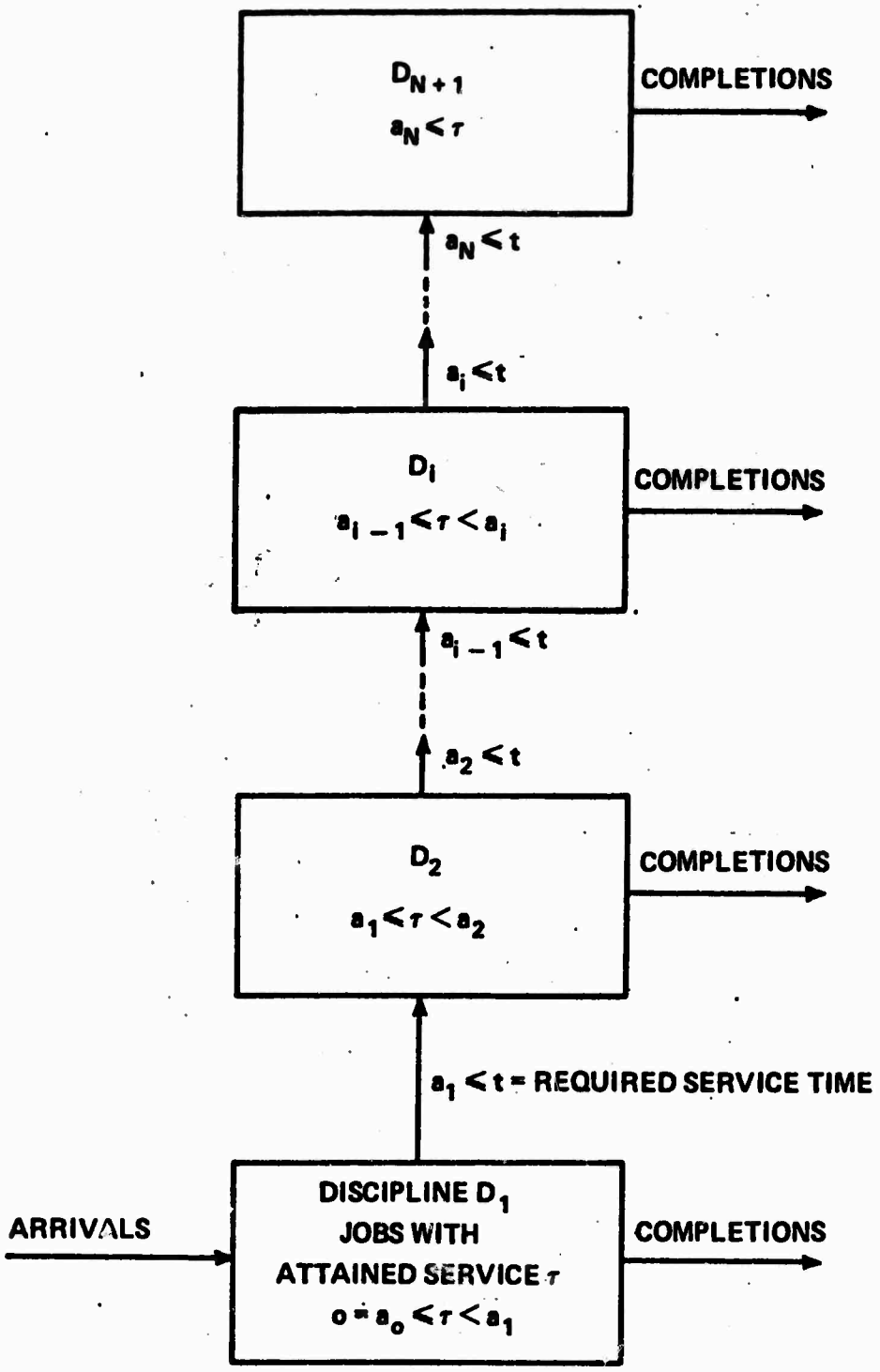


Figure 2.2. The Multilevel Queuing Structure with Disciplines, D_i

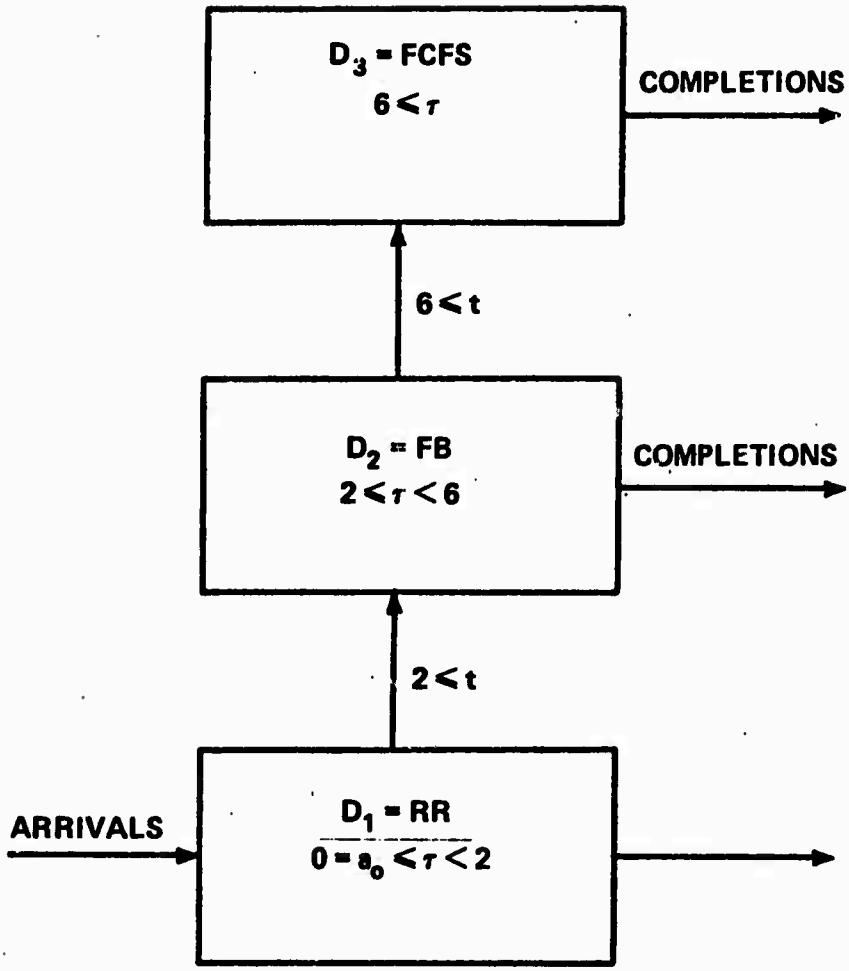


Figure 2.3. Example of N = 2

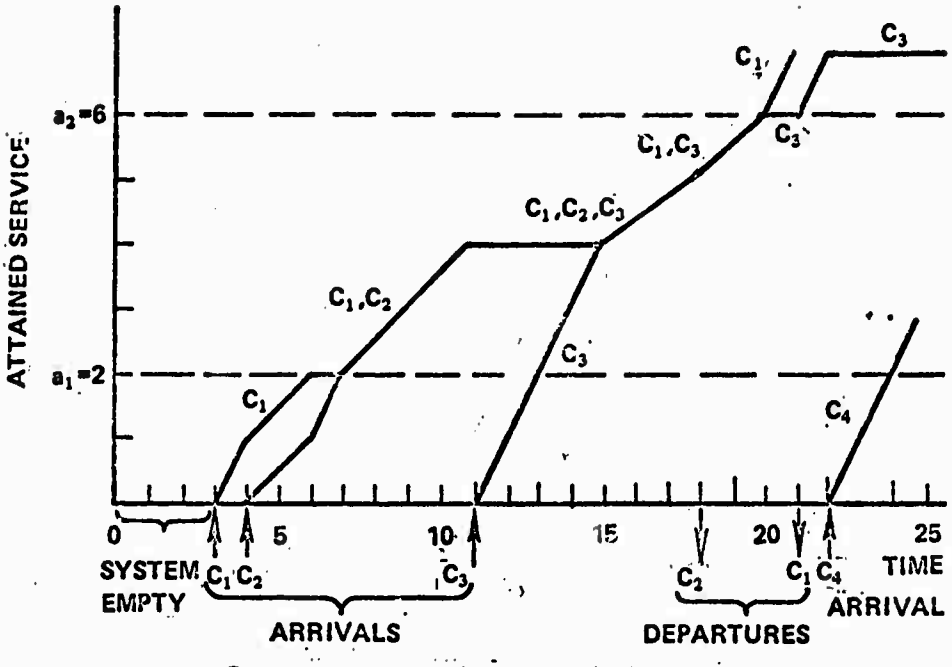
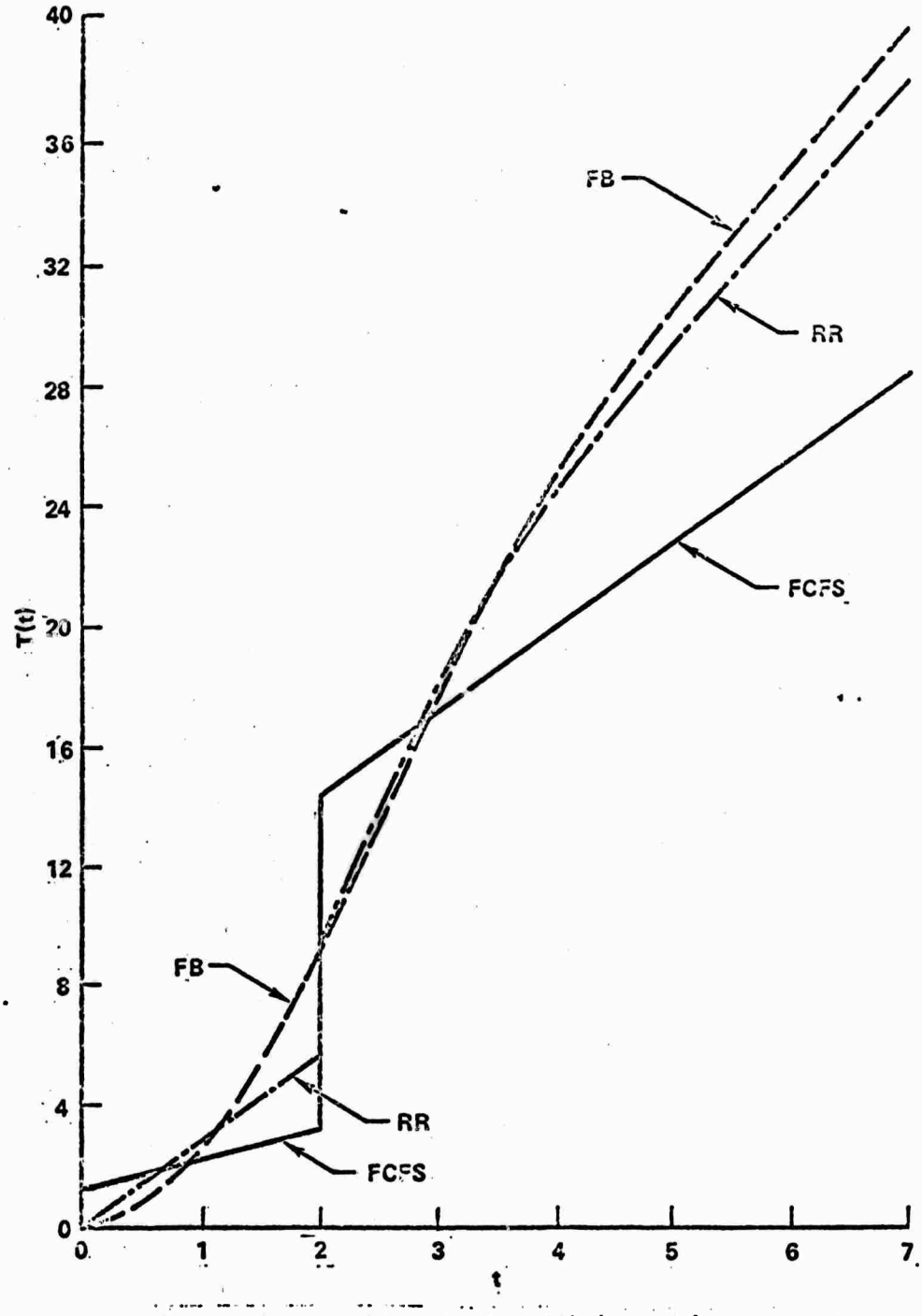


Figure 2.4. History of Customers in Example



Response Time Possibilities for N = 1, M/M/1, $\mu = 1$, $\lambda = .75$, $a_1 = 2$

Fig. 4.1

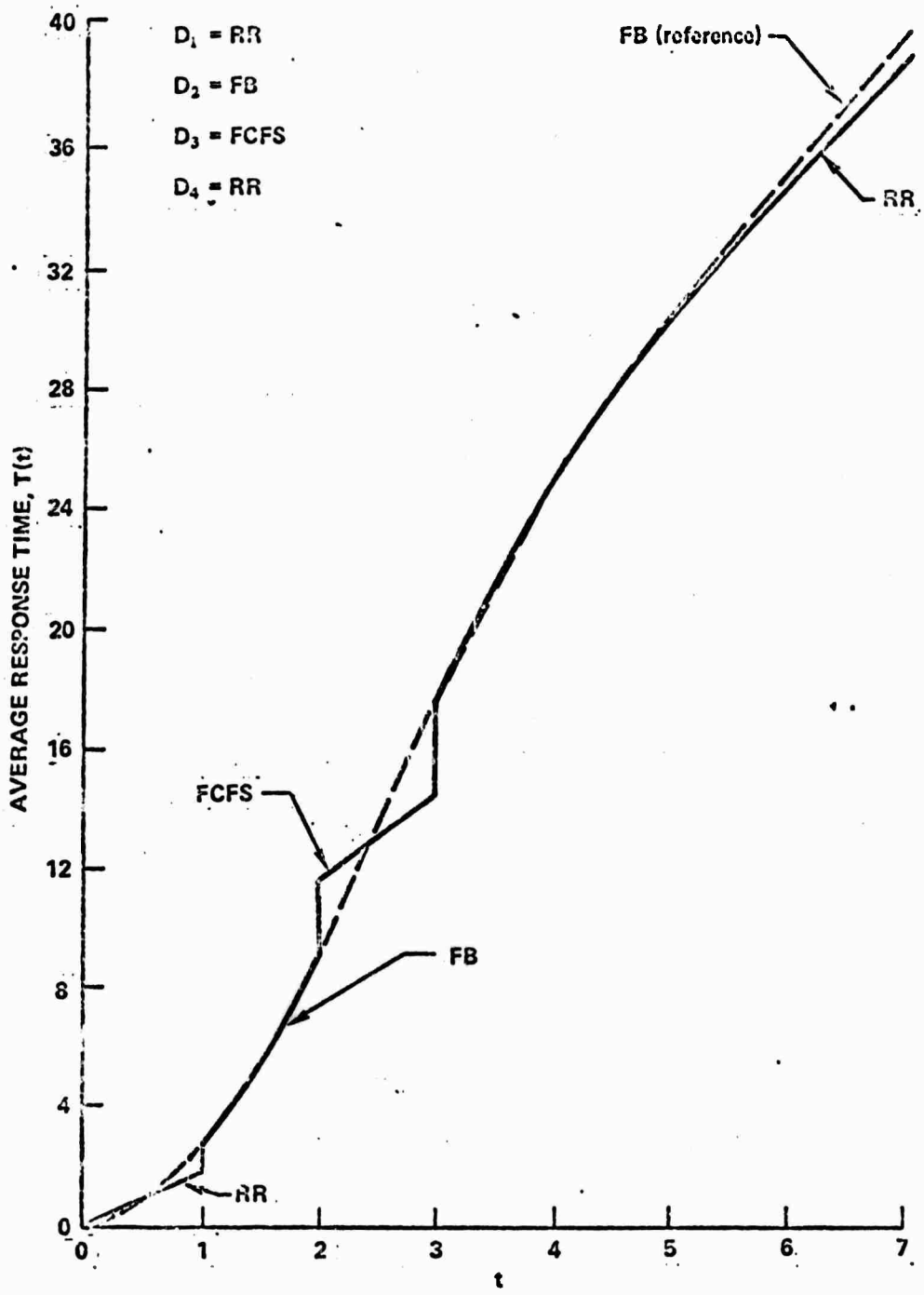


Figure 4.2. Response Time for an Example of $\bar{N} = \bar{3}$, M/M/1, $\mu = 1$, $\lambda = 0.75$, $a_i = i$

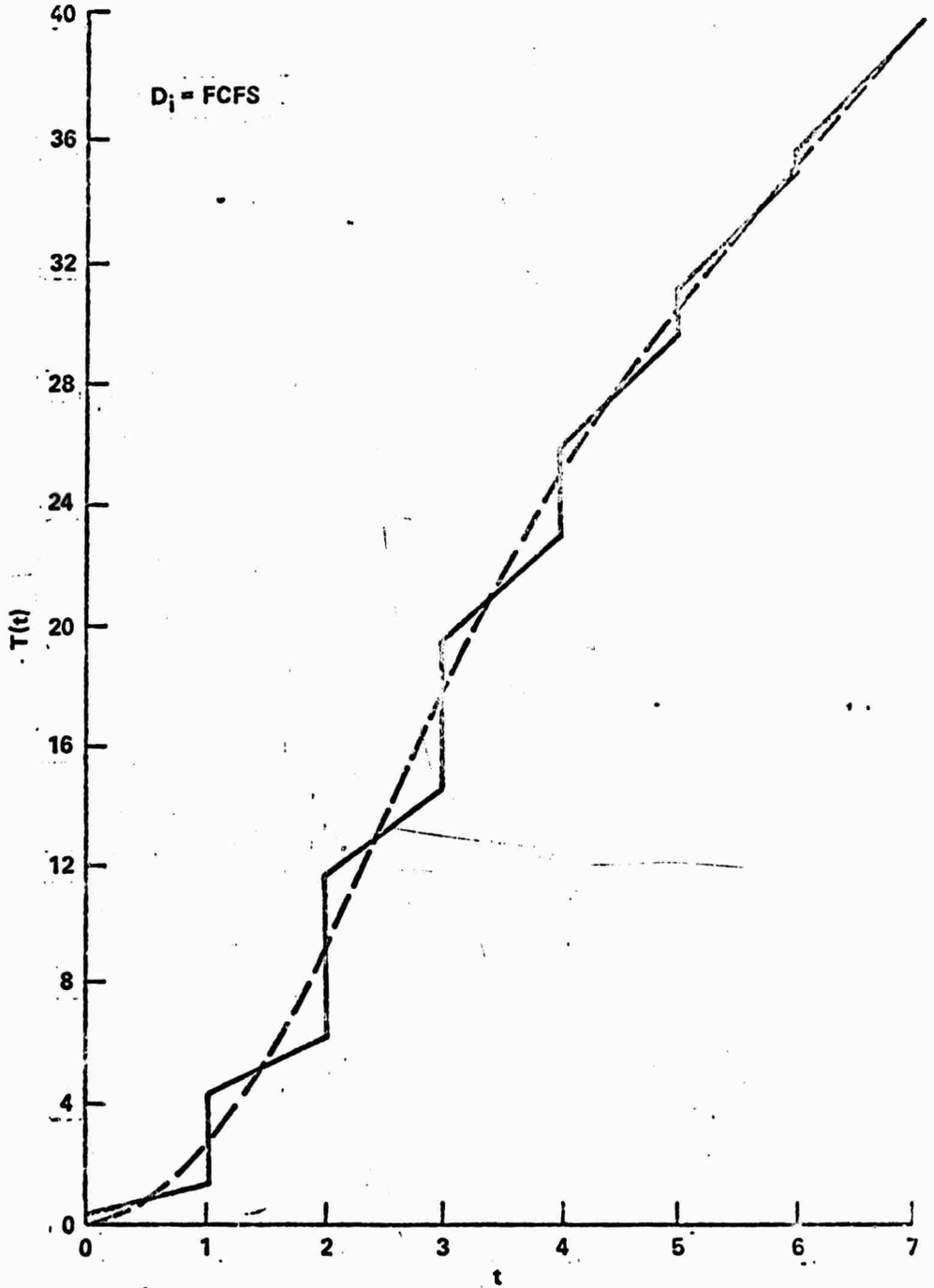


Fig. 4. 3. Response Time for the M/M/1 Iterated Structure, $\mu = 1, \lambda = 0.75, a_i = i, N = \infty$

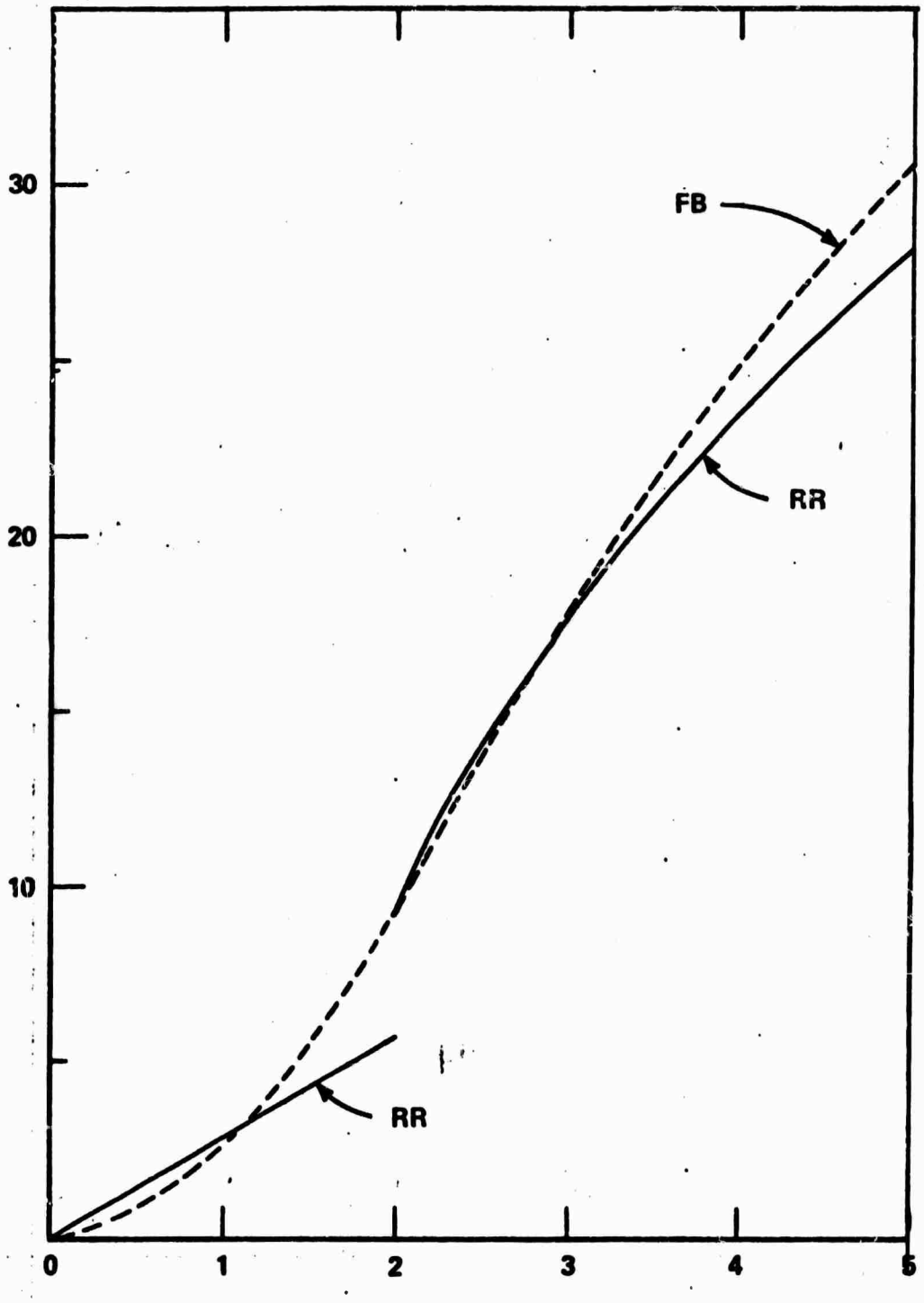


Figure 4.4. Response Time for Example of D_i = RR, M/M/1, $\mu = 1.0, \lambda = 0.75, s_1 = 2.0, s_2 = 5.0$

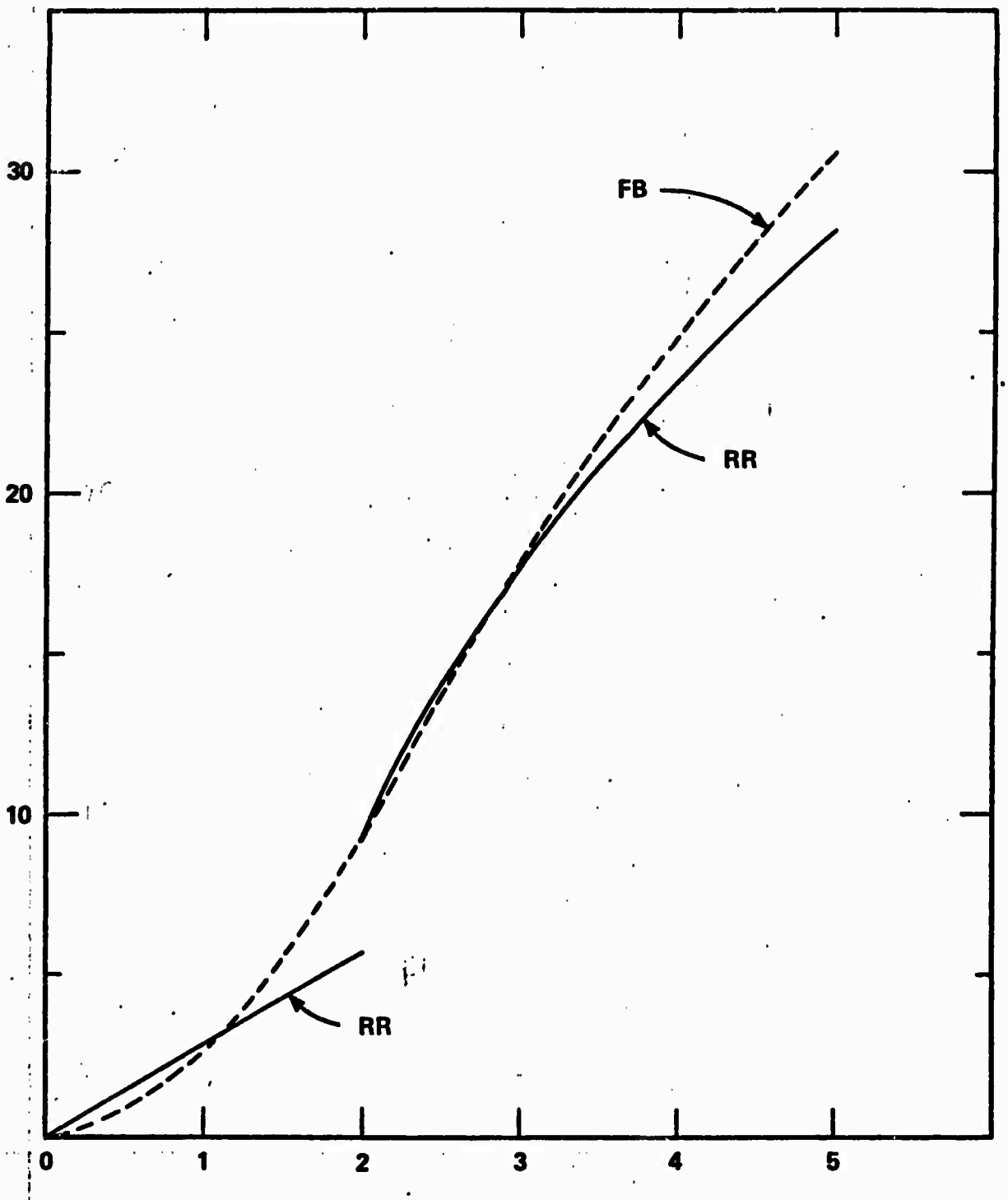


Figure 4.4. Response Time for Example of D_i = RR, M/M/1, $\mu = 1.0, \lambda = 0.75, a_1 = 2.0, a_2 = 5.0$

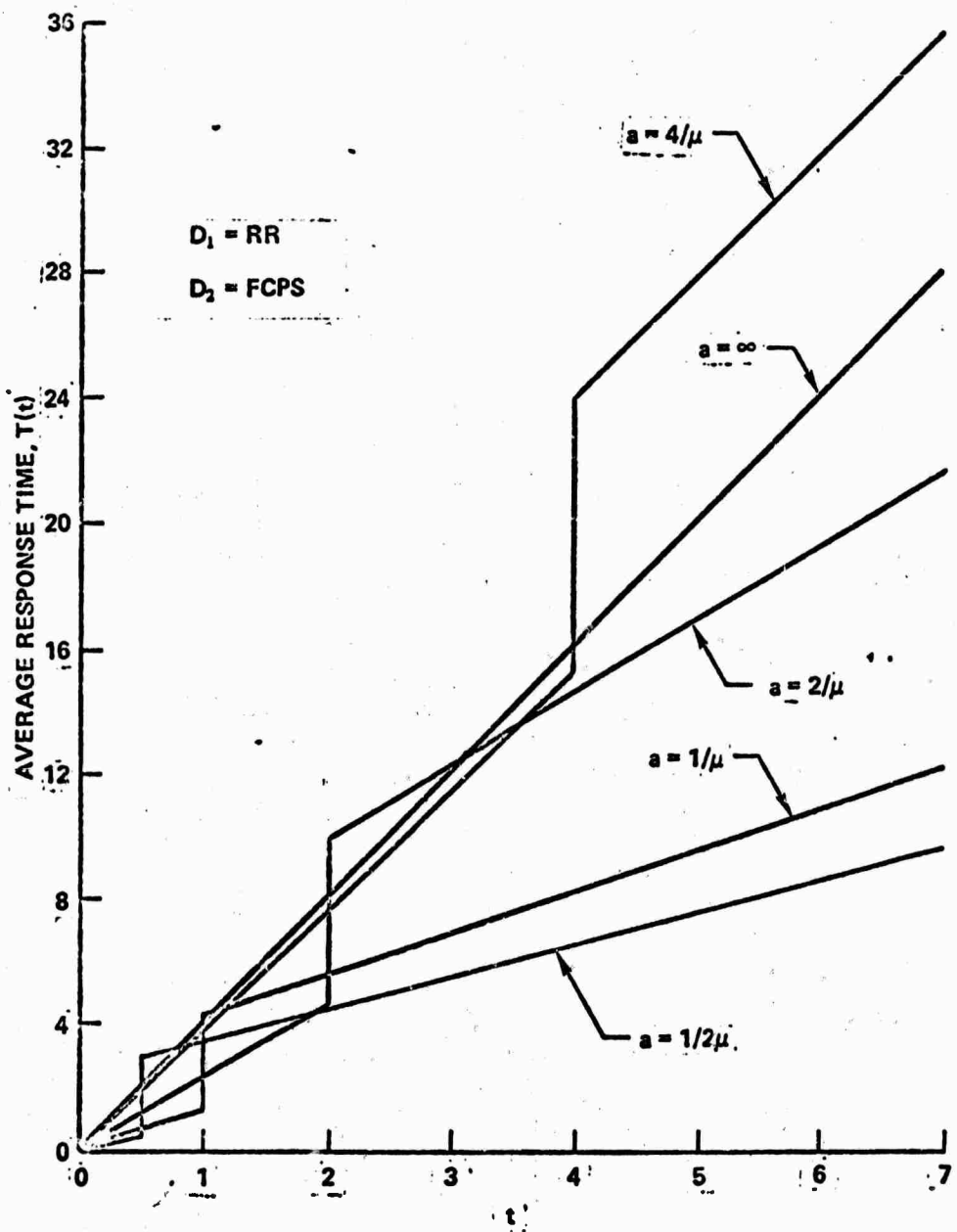


Figure 4.5. Response Time for RR, FCFS in $M/E_2/1$ with $\mu = 1$, λ 0.75 and $a = 1/2, 1, 2, 4, \infty$

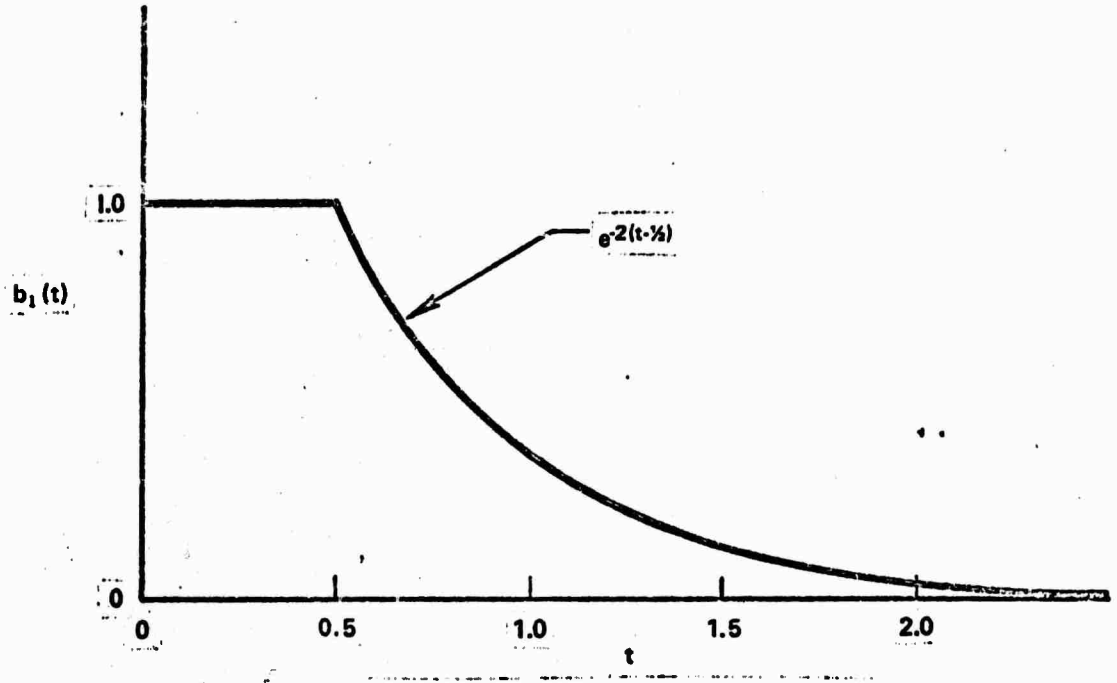


Fig 4.6. EXAMPLE SERVICE TIME DENSITY, $b_1(t)$

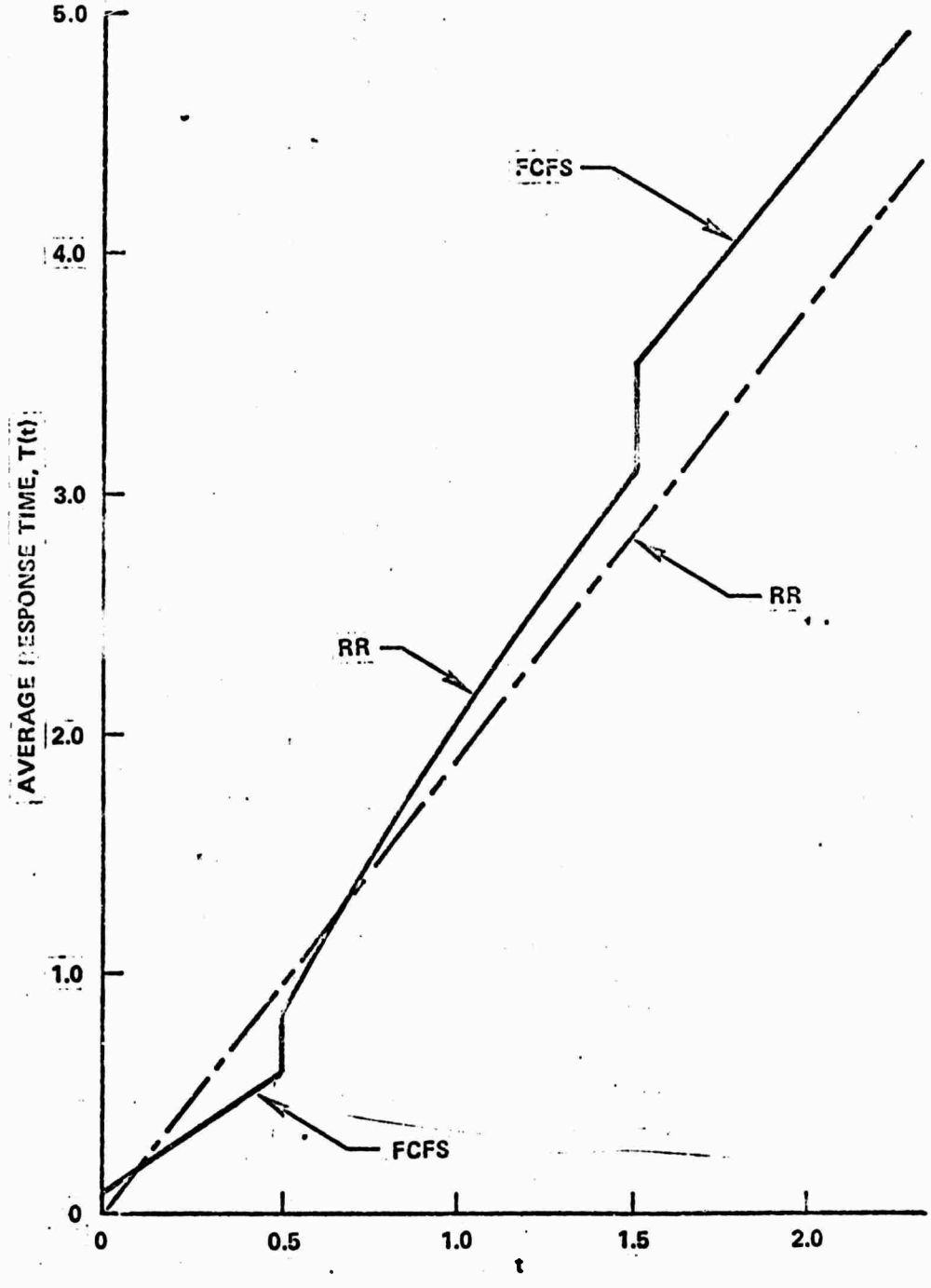


Fig. 4.7. RESPONSE TIME FOR AN EXAMPLE OF N = 2, λ = 0.75, SERVICE TIME DENSITY = $b_1(t)$

APPENDIX C

TIGHT BOUNDS ON THE AVERAGE RESPONSE TIME FOR TIME-SHARED COMPUTER SYSTEMS

by L. Kleinrock, R. R. Muntz and J. Hsu

TIGHT BOUNDS ON THE AVERAGE RESPONSE TIME FOR TIME-SHARED COMPUTER SYSTEMS*

LEONARD KLEINROCK, RICHARD R. MUNTZ, and JIUNN HSU**
Computer Science Department
University of California, Los Angeles, California, U.S.A.

In this paper, some fundamental properties are established which apply to the average response time functions for all time-shared computer systems. The first property is one of monotonicity. The second is a conservation law which provides insight into the trade-offs available as one varies the response time function by changing the scheduling algorithm.

The main thrust of the paper is to establish tight upper and lower bounds on the average response time. All these equilibrium results are good for Poisson arrivals, arbitrary service time distribution and arbitrary (but work-conserving) scheduling algorithms which can take advantage only of arrival time and attained service time. Examples of these properties are given for a number of service-time distributions and scheduling algorithms.

1. INTRODUCTION

We are in the midst of a veritable explosion regarding the number of published papers which give analytical results for computer systems! This seems especially true in the modeling and analysis of time-shared computer systems.[1]

It is fair to say that the recognition of probabilistic models as the appropriate method for studying these systems was that which permitted the breakthrough in analysis. In particular, the use of queueing theory has been most profitable in this analytic work.

As a result of this flood of results, each applying to a slightly different set of assumptions, it is natural that we should seek some order in this embarrassment of riches. For example, do there exist any invariants in behavior? Can we bound the possible range of performance, regardless of structure? What constitutes feasible performance profiles for these systems? These, and many more, are reasonable inquiries to make amidst the confusion of results.

In this paper we adopt the point of view that such questions are important and must be answered. Our focus is on a class of models for time-shared computer systems. For these systems we are able to state a monotonicity property, a conservation law, and tight upper and lower bounds on the system performance as measured by average response time.

It is worthwhile mentioning that numerous papers have recently been published which address themselves to bounds, inequalities and approximate solutions to general queueing systems. Among these are Marshall [2,3], Kingman [4], Iglehart [5], Daley and Moran [6], and Gaver [7] to mention a few.

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285).

**This author (J.H.) wishes to acknowledge his gratitude to the International Business Machine Corporation for the granting of an IBM Fellowship.

[1] See, for example, the recent survey by McKinney [1].

2. THE CLASS OF SYSTEMS

Our objective is to create some order among many of the results available in the analysis of time-shared computer systems. Let us consider the class of systems described below.

We adopt the well-known [8] feedback queueing model for time-shared systems shown in Fig. 2.1.

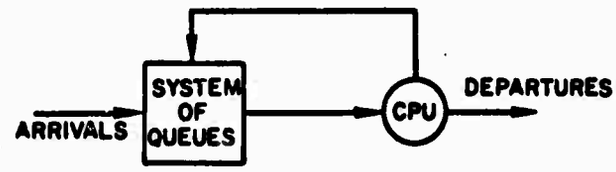


Fig. 2.1. General Feedback Queueing Model

In this model it is assumed that the central processing unit (CPU) is the only resource being accessed. Jobs arrive according to a Poisson process with an average arrival rate λ jobs/sec. They each bring a demand for service by the CPU in an amount equal to t seconds, where these demands are chosen independently from the service time distribution $B(t)$:

$$B(t) = P[\text{service time} \leq t \text{ seconds}] \qquad (2.1)$$

We define the usual moments of service time as[2]

$$\overline{t^n} = E[t^n] = \int_0^\infty t^n dB(t) \qquad (2.2)$$

[2] Where E denotes the expectation operator.

We further define the utilization factor[3]

$$\rho = \lambda \overline{t} \qquad (2.3)$$

Upon arrival, a job enters the system of queues where he waits for a "turn" at service. When, finally, his turn comes up, he is provided a quantum of service equal to q seconds. If he requires less than (or equal to) q seconds, he departs upon completion; if not, he returns to the system of queues having been partially served, in which case we say that he has an attained service of q seconds. Eventually, he will be permitted a second quantum, etc., finally leaving when his total attained service equals his required service time. We assume that no overhead (in time) is incurred in transferring customers in and out of service (i.e., no loss or swap-time); it is possible to account for swap-time [9] in these models, but we do not pursue that matter here.

The decision rule which chooses the next customer to receive a quantum is referred to as the scheduling algorithm. We assume that the scheduling algorithm makes use only of $\lambda, B(t)$, a job's arrival time and a job's attained service.

In this paper, we consider a very useful special case of the above model in which we permit the quantum q to approach zero. This limit is known as the processor-sharing model [10] for time-shared systems. In this case, our model in Fig. 2.1 becomes that of Fig. 2.2 in which more than one customer (say n) may be sharing the processor simultaneously; in such a case each customer receives service at a rate of $1/n$ seconds of service/second.

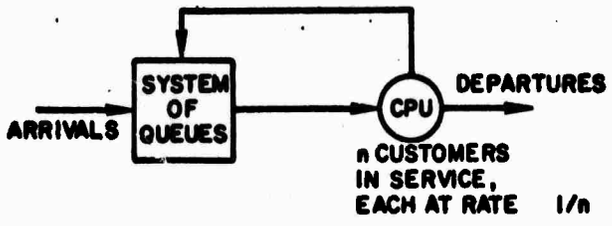


Fig. 2.2. Feedback Queueing Model for Processor Sharing

Response time is the interval measured from when a customer arrives demanding service until he departs fully serviced. For a customer requiring t seconds of service, the average response time is denoted.[4]

$$T(t) = \text{average response time for customer requiring } t \text{ seconds of service} \qquad (2.4)$$

This quantity is usually taken as the measure of performance for time-shared systems for good reason. In particular, it is usually desired that short jobs (small t) be given preferential treatment over long jobs; this discriminatory performance is easily seen through the function $T(t)$.

A function closely related to the average response time $T(t)$, is the average wasted or waiting time $W(t)$ defined as

$$W(t) = T(t) - t \qquad (2.5)$$

Furthermore, we consider a third related function, $W(t)/t$ which may be interpreted as the penalty rate to jobs requiring t seconds of service since it gives the ratio of the cost in time $(W(t))$ which must be paid per second of useful service time (t).

It is convenient to introduce some additional notation at this point. Let us define

$$\overline{t_x^n} = \int_0^x t^n dB(t) + x^n [1 - B(x)] \qquad (2.6)$$

which is the n^{th} moment of the service time distribution if service times are truncated at x seconds. Also let

$$\rho_x = \lambda \overline{t}_x \qquad (2.7)$$

and

$$\overline{W}_x = \frac{\lambda \overline{t_x^2}}{2(1 - \rho_x)} \qquad (2.8)$$

Note that $\overline{t_\infty^n} = \overline{t^n}$, $\rho_\infty = \rho$ and that $\overline{W}_\infty$ is the expected work (backlog) found by a new arrival to the queueing system[5] M/G/1 [11].

In summary then, the class of systems we consider is the class of M/G/1 processor-shared time-sharing systems with zero swap-time and arbitrary scheduling algorithms.

3. RESPONSE TIME FUNCTIONS

From the published literature, we find many results for processor-shared systems. Some of these we describe in this section.

1. Batch-processing, first-come-first-served (FCFS)

   In the FCFS system, the oldest job in the system is given complete use of the CPU until it completes its required service. For these systems, we have [11]

[3] The systems we consider are assumed to be equilibrium, which requires $\rho < 1$.

[4] Since we have $\rho < 1$, we consider steady-state results only, an example of which is $T(t)$.

[5] The notation M/G/1, common in queueing theory, denotes a single server system with Poisson arrival process and arbitrary service time distribution.

$$W(t) = \bar{W}_\infty = \frac{\lambda \overline{t^2}}{2(1-\rho)} \qquad (3.1)$$

2. Round-Robin (RR)

In the RR system, all customers share the CPU equally. We have [10][6]

$$W(t) = \frac{\rho t}{1-\rho} \qquad (3.2)$$

3. Selfish Round-Robin (SRR)

In the SRR system, all customers with the highest value of "priority" share the CPU equally; all others wait in the queue. Priority for a job is calculated as $\alpha w + \beta s$ where $\alpha \geq \beta \geq 0$ are constants and w is the time spent waiting and s is the time spent in the CPU (perhaps shared) for that job. The SRR system has only been solved for exponential service time, i.e.,

$$B(t) = 1 - e^{-\mu t} \qquad t \geq 0 \qquad (3.3)$$

In this case we have [12]

$$W(t) = \bar{W}_\infty + \frac{(t-\bar{t})(1-\beta/\alpha)\rho}{1-\rho(1-\beta/\alpha)} \qquad (3.4)$$

4. Generalized Foreground-Background (FB)

The FB system shares the CPU equally among all those jobs which have the smallest attained service. For the FB system we have [13]

$$W(t) = \frac{\bar{W}_t + t\rho_t}{1-\rho_t} \qquad (3.5)$$

5. Multilevel (ML)

In the ML system, a set of attained service times $\{a_i\}$ is defined such that

$$0 = a_0 \leq a_1 \leq a_2 \leq \cdots \leq a_N \leq a_{N+1} = \infty \qquad (3.6)$$

When a job's attained service falls in the i^{th} interval $[a_{i-1}, a_i)$, then the scheduling algorithm followed for this job is denoted D_i where D_i may be FCFS, RR or FB. The discipline followed between the levels is FB. Results for these ML systems are reported in [14] for arbitrary $B(t)$ (with some additional restrictions on $B(t)$ when $D_i = RR$ for $i \geq 2$).

Note that the FCFS system offers no discrimination based on attained service time, whereas the FB system discriminates as much as possible on this basis. The RR system is "fair" in the sense that the penalty rate $W(t)/t$ is independent of service time, t.

For these processor-shared systems, it is useful to display, in one figure, the wasted time $W(t)$. This we do in Fig. 3.1 for the case of exponential service (see Eq. (3.3)) with $\lambda = 0.75$ and $\bar{t} = 1.0$ (thus $\rho = 0.75$). We purposely superimpose the performance curves for many scheduling disciplines. We are confronted with quite a selection of possible performance functions!

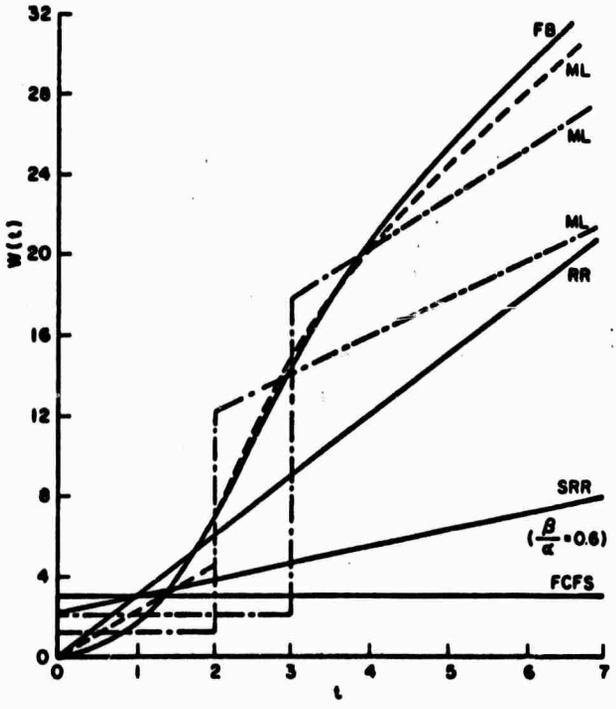


Fig. 3.1. A Set of Response Curves for M/M/1, $\bar{t} = 1.0$, $\lambda = 0.75$, $\rho = 0.75$.

One might naturally inquire as to whether these curves are confined to any particular region in the $(W(t),t)$ plane. The answer is definitely yes,[7] and we develop these and other constraints in the next section.

4. RESULTS

In this section we present results concerning the response functions $(W(t))$ which are feasible when the scheduling discipline is based only on attained service times and elapsed waiting times of jobs. In Section 4.1 below we describe several fundamental characteristics of $W(t)$ and, in particular, we give a conservation relationship which the response function must satisfy. In Sections 4.2 and 4.3, tight lower and upper bounds are derived for response functions in the sense that for any $W(t)$, $W_\ell(t) \leq W(t) \leq W_u(t)$.

[6] It is also true that we obtain the identical $W(t)$ for the last-come-first-served system (LCFS).

[7] In fact, if the reader looks at this figure and squints his eyes, he can almost guess the shape of such bounds.

4.1. A Monotonicity Property and a Conservation Law for $W(t)$

We are considering scheduling disciplines in which each job is characterized by (1) its attained service time, t_s and (2) its elapsed waiting time, t_w. Therefore, the state of the system is the number of jobs in the system and t_s and t_w for each job. A particular scheduling discipline may effectively ignore one or both of these parameters, but this information is assumed to be available for each job. Because scheduling decisions are made only on the basis of these two parameters, the following statement is self-evident. The history of a job requiring $t_1 \geq t$ seconds of service from the time of its arrival at the system until it has received t seconds of service is independent of the exact value of t_1. A direct consequence of this fact is that $W(t)$ is a non-decreasing function or equivalently

$$W'(t) \equiv \frac{dW(t)}{dt} \geq 0 \qquad (4.1)$$

In deriving $W_\ell(t)$ and $W_u(t)$ we shall need another result which is given below. From [8] we have that

$$n(t) = \lambda[1 - B(t)][W'(t) + 1] \qquad (4.2)$$

where $n(t)$ is the density of jobs in the system with t seconds of attained service time. We define the "work" in the system at time t as the additional time required to empty the system if no new arrivals are permitted entry; this is also referred to as the "unfinished work" and as the "virtual waiting time." The mean work $\bar{W}$ in the system can be expressed as

$$\bar{W} = \int_{0^-}^{\infty} n(t)\, E[\text{remaining service time for a job with attained service time of } t]\, dt$$

or $$\bar{W} = \int_{0^-}^{\infty} n(t) \int_{t}^{\infty} (\tau - t)\, \frac{dB(\tau)}{1 - B(t)}\, dt$$

Substituting from (4.2)

$$\bar{W} = \lambda \int_{0^-}^{\infty} (\bar{W}'(t) + 1) \int_{0}^{\infty} (\tau - t)\, dB(\tau)\, dt$$

By changing the order of integration

$$\bar{W} = \lambda \int_{0}^{\infty} \left[\int_{0^-}^{\tau} (W'(t) + 1)(\tau - t)\, dt \right] dB(\tau) \qquad (4.3)$$

Integrating the inner integral by parts

$$\int_{0^-}^{\tau} (W'(t) + 1)(\tau - t)\, dt$$

$$= (\tau - t)(W(t) + t)\Big|_{0^-}^{\tau} + \int_{0^-}^{\tau} [W(t) + t]\, dt$$

$$= \int_{0^-}^{\tau} [W(t) + t]\, dt$$

Substituting into Eq. (4.3)

$$\bar{W} = \lambda \int_{0}^{\infty} \int_{0^-}^{\tau} [W(t) + t]\, dt\, dB(\tau)$$

Again changing the order of integration

$$\bar{W} = \lambda \int_{0^-}^{\infty} [W(t) + t] \int_{t}^{\infty} dB(\tau)\, dt$$

$$= \lambda \int_{0}^{\infty} [W(t) + t][1 - B(t)]\, dt$$

But in general, we have that

$$\int_{0}^{\infty} t[1 - B(t)]\, dt = \frac{\overline{t^2}}{2}$$

Moreover, the mean work in the system is known [11] to be

$$\bar{W} = \frac{\lambda \overline{t^2}}{2(1 - \rho)} \qquad (4.4)$$

Thus we have the following conservation laws for $T(t)$ and $W(t)$:

$$\frac{\overline{t^2}}{2(1 - \rho)} = \int_{0}^{\infty} T(t)[1 - B(t)]\, dt \qquad (4.5)$$

and

$$\frac{\rho \overline{t^2}}{2(1 - \rho)} = \int_{0}^{\infty} W(t)[1 - B(t)]\, dt \qquad (4.6)$$

We refer to Eqs. (4.5) and (4.6) as Conservation Laws since they are based on the conservation of average unfinished work in the system. This places an integral constraint on $W(t)$ (and $T(t)$) as a second necessary condition, regardless of

scheduling discipline. The implications of the conservation law may be seen by recognizing that $[1 - B(t)]$ is a nonincreasing function of t. Thus, if one had a given $W(t)$ as a result of some scheduling algorithms, and then changed the algorithm so as to reduce $W(t)$ over some interval $(0,t_0)$, then the conservation law would require that the new $W(t)$ be considerably above the old value for some range above t_0. This follows since the weighting factor, $1 - B(t)$, is smaller for larger t.

4.2. The Lower Bound

We claim that to minimize $W(x)$ the scheduling discipline must

1. never service jobs with attained service time greater than or equal to x while there are jobs in the system with attained service time less than x, and
2. never preëmpt a job once it has been selected for service until it has at least x seconds of attained service time.

Under these conditions the response function in the interval $(0,x)$ is just the response function for a nonpreëmptive system with service times truncated at x. For convenience we will assume a FCFS scheduling discipline. In this case the response function (denoted $W_{FCFS-x}(t)$) has the form shown in Fig. 4.1 (see, for example, [14]). Note that $W'_{FCFS-x}(t) = 0$ over $(0,x)$. The scheduling of jobs with attained service time greater than x is of no concern in this argument as long as condition 1 is maintained.

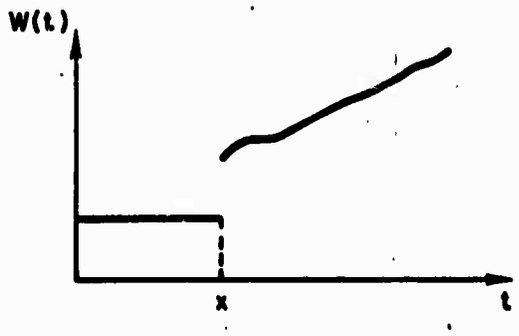


Fig. 4.1. Response for FCFS up to x Seconds of Service

Let $\bar{W}_x$ be the mean work in the system excluding work to be done on jobs beyond providing x seconds of attained service to each. In other words, if a job requires $t > x$ seconds of service and has received $y < x$ seconds of service, its contribution to $\bar{W}_x$ is $x - y$. By the same method used to derive Eq. (4.5) it can be shown that

$$\bar{W}_x = \lambda \int_0^x [W(t) + t][1 - B(t)]dt$$

Now since $W_{FCFS-x}(t)$ has minimum slope (i.e., 0) over the interval $(0,x)$, and due to the monotonicity given in Eq. (4.1), if any other response curve $W(t)$ is such that $W(x) < W_{FCFS-x}(x)$ it must be such that $W(t) < W_{FCFS-x}(t)$ for $0 \le t \le x$. But under condition 1 above, $\bar{W}_x$ has its minimum value since work in this class is continuously decreased at maximum rate whenever there is such work in the system. Therefore, for any $W(t)$,

$$\lambda \int_0^x [W(t) + t][1 - B(t)dt$$

$$\ge \lambda \int_0^x [W_{FCFS-x}(t) + t][1 - B(t)]dt$$

Thus we conclude that $W(t) < W_{FCFS-x}(t)$ in $(0,x)$ is impossible and therefore $W(x) \ge W_{FCFS-x}(x)$.

The lower bound $W_\ell(t)$ is given by the waiting time for the FCFS discipline with the service times truncated at t, namely [14]

$$W_\ell(t) = \frac{\lambda \overline{t_t^2}}{2(1 - \rho_t)} \qquad (4.7)$$

Note that $W_\ell(0) = 0$ and that $W_\ell(\infty) = \bar{W}$; also $W'_\ell(0) = W'_\ell(\infty) = 0$.

4.3. The Upper Bound

In this case we begin with a discrete time system.

Assume that the service time distribution is of the form

$$\Pr[\text{service time} = kq] = p_k \qquad k = 1,2,3,\ldots$$

where q is the quantum as discussed in Section 2. Therefore, the only possible service time requirements are multiples of q. We shall also assume that arrivals may take place only during the instant before the end of a quantum and that the processor is assigned to a job for a quantum at a time. The probability that an arrival takes place at the end of a quantum is λq so that the mean arrival rate is λ. It should be clear that any continuous service time distribution can be approximated arbitrarily closely by a discrete time distribution by letting q approach 0. Also, these restrictions on the service discipline and arrival mechanism are effectively eliminated when $q \to 0$. In this discrete time model our goal is to maximize $W(kq)$.

We claim that the following scheduling rule is necessary and sufficient to maximize $W(kq)$: no allocation of a k^{th} quantum is made to any job when there is some other job in the system waiting for its j^{th} quantum where $j \neq k$. We note in passing that many scheduling disciplines will satisfy this rule.

We relabel the time axis so that $t = 0$ at an arbitrary point in some idle period. The times at which some job is allocated a k^{th} quantum we call "critical times." Let c_i be the time that the i^{th} critical time occurs. We wish to maximize $\bar{c}_\ell$ (the average of c_ℓ) for some fixed ℓ, and we will show that to accomplish this it is necessary and sufficient to satisfy the condition that at the ℓ^{th} critical time no job is waiting for a j^{th} quantum where $j \neq k$. Certainly this condition is necessary since if a proposed scheduling discipline did not have this property then c_ℓ can easily be increased when the condition is not satisfied as follows: follow the proposed schedule until the point where the ℓ^{th} critical time would occur and then assign a quantum to a job waiting for its j^{th} $(\neq k)$ quantum.

Since we have already shown necessity, to prove the sufficiency of the condition for maximizing $\bar{c}_\ell$, we need only show that any schedule satisfying the condition yields the same value for $\bar{c}_\ell$. Let A be any scheduling algorithm which satisfies the rule that at the ℓ^{th} critical time no job is waiting for a j^{th} quantum $j \neq k$. Let a_ℓ be the time at which the ℓ^{th} job arrives which will require at least kq seconds of service. The state of the system at a_ℓ will, in general, depend on the algorithm A. In particular, the number of critical times that have occurred prior to a_ℓ (let this be s) is a function of A. Let $E_A[c_\ell - a_\ell | \text{state of system at } a_\ell]$ be the expected value of $c_\ell - a_\ell$ under algorithm A conditioned on the state of the system at a_ℓ. The state of the system is given by the number of jobs in the system, the attained service time of each job in the system and s, the number of critical times that have occurred. Thus, we have

$$\begin{aligned} E_A[c_\ell - a_\ell | \text{state of systems at } a_\ell] &= E_A[\text{remaining work in system not requiring a } k^{th} \text{ quantum} | \text{state of system at } a_\ell] \\ &+ (\ell - s - 1)E[\text{remaining service time for job with } (k-1)q \text{ seconds of attained service}] \\ &+ (k-1)q \\ &+ \lambda\bar{t}_{(k-1)q} E_A[c_\ell - a_\ell | \text{state of the system at } a_\ell] \end{aligned} \tag{4.8}$$

But the sum of the first two terms on the right-hand side of this equation is equal to the expected amount of work in the system at a_ℓ given the state at a_ℓ. Thus

$$\begin{aligned} E_A[c_\ell - a_\ell | \text{state of system at } a_\ell] &= E_A[\text{work in system at } a_\ell | \text{state at } a_\ell] \\ &+ (k-1)q \\ &+ \lambda\bar{t}_{(k-1)q} E_A[c_\ell - a_\ell | \text{state of system at } a_\ell] \end{aligned}$$

Removing the condition on the state of the system at a_ℓ we have

$$E_A[c_\ell - a_\ell] = E_A[\text{work in the system at } a_\ell] + (k-1)q + \lambda\bar{t}_{(k-1)q} E_A[c_\ell - a_\ell]$$

$$\text{or } E_A[c_\ell - a_\ell] = \frac{E_A[\text{work in system at } a_\ell] + (k-1)q}{1 - \lambda\bar{t}_{(k-1)q}}$$

But $E_A[\text{work in system at } a_\ell]$ is not a function of the particular scheduling algorithm and therefore $E_A[c_\ell - a_\ell]$ does not depend on A. Since $E[c_\ell] = E[c_\ell - a_\ell] + E[a_\ell]$ and the right-hand side is independent of A, $E[c_\ell]$ is independent of A. Note that the form of Eq. (4.8) depended on A having the property that at c_ℓ there are no jobs in the system waiting for a j^{th} quantum where $j \neq k$. We have now shown that this condition is necessary and sufficient to maximize $E[c_\ell] (= \bar{c}_\ell)$.

We now show that the general scheduling rule to maximize $W(kq)$ is the same rule which maximizes $\bar{c}_\ell$ applied for all ℓ. We have

$$W(kq) = \lim_{n\to\infty} \frac{\sum_{\ell=1}^{n} \bar{c}_\ell - \sum_{\ell=1}^{n} \bar{a}_\ell}{n} \tag{4.9}$$

The $\bar{a}_\ell$ are independent of the scheduling discipline and the proposed scheduling rule is necessary and sufficient to individually maximize the $\bar{c}_\ell$. Therefore, the same rule is necessary and sufficient to maximize $W(kq)$, which establishes our earlier claim.

It should be clear that in a continuous time system we can approach the maximum of $W(x)$ by the following rule: no job with attained service time in the open interval $(x - \epsilon, x)$ (for $\epsilon > 0$) is serviced while there is a job waiting for service which has attained service time outside this interval. By permitting ϵ to shrink to zero, we approach the maximum for $W(x)$.

One scheduling discipline which maximizes $W(x)$ is the two-level system in which jobs are served FCFS in the first level up to x^- seconds of attained service. A job which does not finish is placed in the second level queue. The second queue is serviced FCFS to completion. The second queue has a lower priority and is only serviced when the first queue is empty (see the ML systems described in Section 3). This queueing system satisfies the condition for maximizing $W(x)$ and therefore from [14] we have

$$W_u(t) = \frac{\lambda\overline{t^2}}{2(1-\rho_t)(1-\rho)} + \frac{t\rho_t}{1-\rho_t} \tag{4.10}$$

Note that $W_u(0) = \bar{W} = W_\ell(\infty)$, that $W_u'(0) = 0$, and that $W_u'(\infty) = \rho/(1-\rho)$.

4.4. Summary

In this section we have established the following for any scheduling algorithm

$$T'(t) \geq 1 \tag{4.11}$$

$$W'(t) \geq 0 \tag{4.12}$$

$$\int_0^\infty T(t)[1 - B(t)]dt = \frac{\overline{t^2}}{2(1-\rho)} \tag{4.13}$$

$$\int_0^\infty W(t)[1 - B(t)]dt = \frac{\rho\overline{t^2}}{2(1-\rho)} \tag{4.14}$$

$$\frac{\lambda\overline{t_t^2}}{2(1-\rho_t)} \leq W(t) \leq \frac{\lambda\overline{t^2}}{2(1-\rho_t)(1-\rho)} + \frac{t\rho_t}{1-\rho_t} \tag{4.15}$$

5. EXAMPLES

Four examples are given in this section to demonstrate the nature of the tight bounds we have obtained. As a performance measure, the equilibrium average waiting times, $W(t)$, are plotted as a function of t. We begin with the M/M/1 system (i.e., Poisson arrivals and exponential service). The response functions of Fig. 3.1 are given again in Fig. 5.1 with the upper and lower bounds superimposed. At $t = 0$, the upper bound and FCFS start at the same point because, under the constraint of the conservation law, no other scheduling algorithm can give longer average waiting time at $t = 0$ than FCFS. The upper bound approaches the FB response asymptotically as t approaches infinity; therefore, a customer with a very long requested service time (as compared to the mean) cannot be delayed much more than he is with FB. The lower bound starts at zero (as does the FB curve), increasing less rapidly with t than the upper bound. It approaches the FCFS curve asymptotically as t goes to infinity. Thus we note that the least discriminating scheduling algorithm (FCFS) touches the upper bound at $t = 0$ and forms the asymptote for the lower bound as t approaches infinity; conversely, the most discriminating scheduling algorithm (FB) touches the lower bound at $t = 0$ and forms the asymptote for the upper bound as t approaches infinity. The above-mentioned behavior of the upper and lower bounds applies not only for the M/M/1 system, but also holds true for any M/G/1 system in general, although the rate of convergence for the bounds to their respective limits varies for different service distributions.

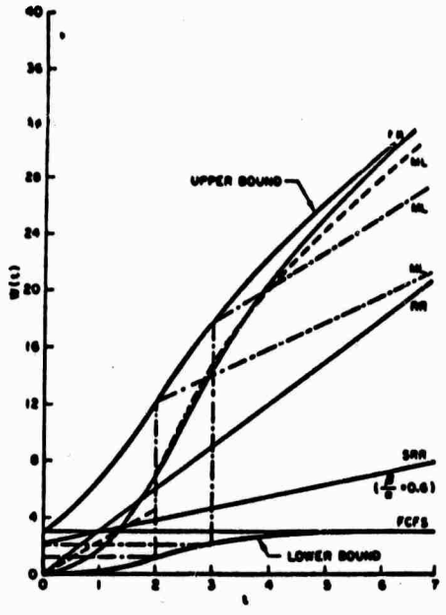


Fig. 5.1. Bounds on Response for M/M/1, $\bar{t} = 1.0$, $\lambda = 0.75$, $\rho = 0.75$.

For the second example we choose the system $M/E_2/1$. In this system we have

$$\frac{dB(x)}{dx} = (2\mu)^2 x e^{-2\mu x} \qquad x \geq 0 \tag{5.1}$$

with mean service time equal to $1/\mu$; the second moment of this distribution is $3/2\mu^2$. Because the second moment is smaller than that of the exponential distribution (whose value is $2/\mu^2$), the bounds are tighter in this example than the M/M/1 case, just as one would expect. Fig. 5.2 shows the behavior of this system with $\mu = 1$ and $\lambda = 0.75$. It is obvious from the figure that for $t > 5/\mu$, the upper and lower bounds have essentially reached their asymptotic form.

In the third example we show the bounds for the $M/H_2/1$ system, where H_2 stands for hyperexponential service distribution with

$$\frac{dB(x)}{dx} = 0.5\mu_1 e^{-\mu_1 x} + 0.5\mu_2 e^{-\mu_2 x} \qquad x \geq 0 \tag{5.2}$$

We choose $\mu_1 = 5\mu$, $\mu_2 = (5/9)\mu$, resulting in a mean service time of $1/\mu$. The second moment of this distribution is $82/25\mu^2$. Fig. 5.3 shows the behavior of the $M/H_2/1$ system with $\mu = 1$ and $\lambda = 0.75$. The upper and lower bounds approach their respective limits at a slower rate than

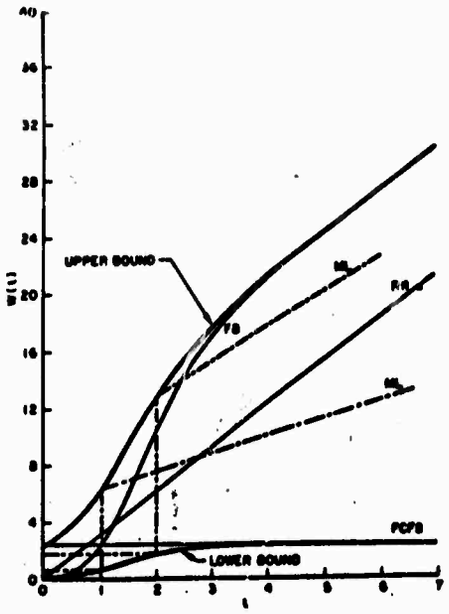


Fig. 5.2. Bounds on Response for $M/E_2/1$, $\bar{t} = 1.0$, $\lambda = 0.75$, $\rho = 0.75$.

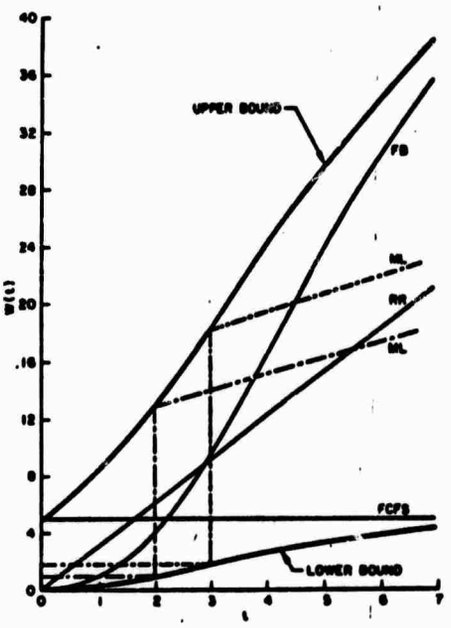


Fig. 5.3. Bounds on Response for $M/H_2/1$, $\bar{t} = 1.0$, $\lambda = 0.75$, $\rho = 0.75$.

either M/M/1 or $M/E_2/1$ because of the larger second moment.

For our last example we choose the system M/U/1 where U stands for uniform service distribution. For this particular example we have

$$\frac{dB(x)}{dx} = \begin{cases} 0.25 & 2 \le x \le 6 \\ 0 & \text{otherwise} \end{cases} \qquad (5.3)$$

and $\lambda = 0.1875$, $\bar{t} = 4.0$, $\rho = 0.75$. Fig. 5.4 shows the behavior of this system. Notice that when $t \ge 6$, the upper bound coincides exactly with the FB curve and that the lower bound coincides exactly with the FCFS curve. The probability of having any customer requesting more than six seconds of service in this example is, of course, equal to zero.

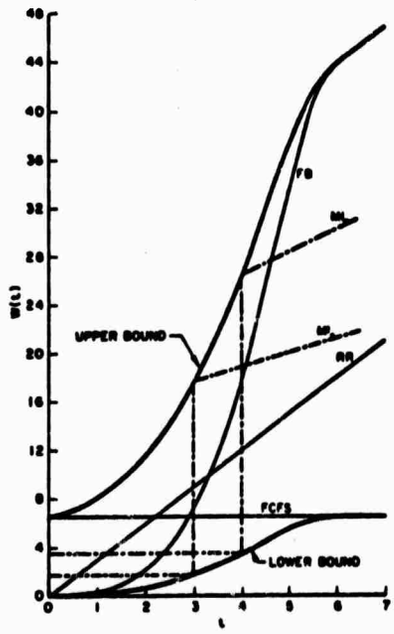


Fig. 5.4. Bounds on Response for M/U/1, $\bar{t} = 4.0$, $\lambda = 0.1875$, $\rho = 0.75$.

Another performance measure, W(t)/t, is given in Fig. 5.5 for the M/M/1 case and is of interest to us, since (as mentioned in Section 2) it gives some feeling for how large a price (in terms of wasted time) a customer must pay in order to get a unit of service time. For the case of RR, this measure is a constant; thus each customer has the

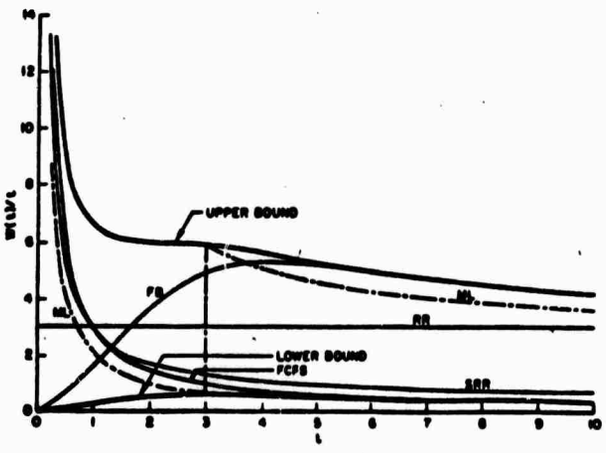


Fig. 5.5. Bounds on Penalty Rate for M/M/1, $\bar{t} = 1.0$, $\lambda = 0.75$, $\rho = 0.75$.

same penalty rate, regardless of his service time. In this sense, everyone is treated equally in the PP System. The curve representing FCFS is monotonically decreasing with t, and so the longer jobs pay at a smaller penalty rate; in this case, system users might attempt to "pool" their requests to take advantage of this "quantity discount." Another extreme example is provided by FB; $W(t)/t$ increases rapidly when t is small, then drops slowly to a constant $(\rho/(1-\rho))$. A customer with a long request can do better by breaking his job into smaller independent jobs and submitting them separately to the system (if this is possible) because then the penalty rate will be greatly reduced.

Fig. 5.6 shows the range of the bounds for the M/M/1 system with $\rho = 0.75$, 0.5 and 0.25, respectively. As can be seen, the region included between the upper and lower bounds for a particular utilization factor ρ depends heavily on ρ; the larger the value of ρ, the greater is the vertical separation between the two bounds, thus allowing larger variation of the mean waiting times for different scheduling algorithms.

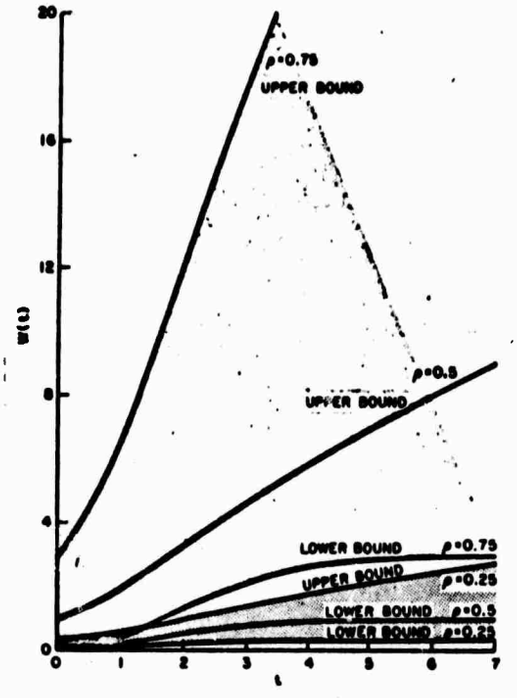


Fig. 5.6. Variation of Bounds for M/M/1 with $\rho = 0.25, 0.50, 0.75$.

6. EXTENSIONS

As we implied in the Introduction, we have answered some fundamental questions regarding the existence of order and structure in the analytical results for time-shared computer systems. Our principal results are given as a monotonicity condition (Eqs. (4.11,4.12)), a conservation law (Eqs. (4.13,4.14)) and tight upper and lower bounds (Eq. (4.15)) on the response function $W(t)$. These results are exemplified by the curves given in Section 5. We note here that although the results were expressed for processor-shared systems, the same type of results apply to the case $q > 0$.

We might observe some additional properties which follow from our results. First we see that any $W(t)$ may touch the lower bound at most once (except over the semi-infinite interval $t_1 \leq t$ when $B(t_1) = 1$); the same may be said for the upper bound.

Secondly, we find that we are able to respond to the following kind of specification. Suppose that a designer requests that all jobs of duration $t \leq t^*$ should have an average wasted time $W(t) \leq W^*$. Then if $W^* \geq W_L(t^*)$, it is possible to guarantee at least this behavior (for example, by an ML system where the first level is FCFS out to t^*). Such a specification seems to us to be quite natural. The next obvious need is to specify the bounds on $W(t)$ which exist for $t > t^*$.

Lastly, we pose the more general question which, at the time of this writing remains unsolved, namely, what are the necessary and sufficient conditions for a given response function to be feasible? This paper has presented some important necessary conditions.

REFERENCES

[1] J. M. McKinney, A Survey of Analytical Time-Sharing Models, Computing Surveys, Vol. 1, No. 2, June 1969, pp. 105-116.

[2] K. T. Marshall, Some Inequalities in Queuing, Operations Research, Vol. 16, No. 3, May-June 1968, pp. 651-665.

[3] K. T. Marshall, Bounds for Some Generalizations of the GI/G/1 Queue, Operations Research, Vol. 16, No. 4, July-August 1968, pp. 841-848.

[4] J. F. C. Kingman, Some Inequalities for the GI/G/1 Queue, Biometrika, Vol. 49, pp. 315-324.

[5] D. L. Iglehart, Diffusion Approximations in Applied Probability, Math. of the Decision Sciences (part 2), Ed. G. B. Dantzig and A. F. Veinott, Jr.; Amer. Math Soc., Providence, R.I., 1968.

[6] D. J. Daley and P. A. P. Moran, Two-Sided Inequalities for Waiting Time and Queue Size Distributions in GI/G/1, Theory of Probability, Vol. XIII, No. 2, 1968, pp. 338-341.

[7] D. P. Gaver, Diffusion Approximations and Models for Certain Congestion Problems, J. of Applied Prob., Vol. 5, 1968, pp. 607-623.

[8] L. Kleinrock and E. Coffman, Distribution of Attained Service in Time-Shared Systems, J. of Computers and Systems Science, Vol. 3, October 1967, pp. 287-298.

[9] L. Kleinrock, Swap Time Considerations in Time-Shared Systems, IEEE Trans. on Computers, June 1970, pp. 534-540.

[10] L. Kleinrock, Time-Shared Systems: A Theoretical Treatment, J. Assoc. Computing Machinery, Vol. 14, No. 2, April 1967, pp. 242-261.

[11] D. R. Cox and W. L. Smith, Queues, Methuen (1961).

[12] L. Kleinrock, A Continuum of Time-Sharing Scheduling Algorithms, Proc. 1970 SJCC, Atlantic City, May 1970, pp. 453-458.

[13] L. E. Schrage, The Queue M/G/1 with Feedback to Lower Priority Queues, Management Science, Vol. 13, No. 7, 1967.

[14] L. Kleinrock and R. R. Muntz, Multilevel Processor-Sharing Queueing Models for Time-Shared Systems," Proc of the 6th ITTC, Munich, Germany, September 1970, pp. 341/1-341/8.

APPENDIX D

ADAPTIVE ROUTING TECHNIQUES FOR STORE-AND-FORWARD COMPUTER-COMMUNICATION NETWORKS

by G. L. Fultz and L. Kleinrock

ADAPTIVE ROUTING TECHNIQUES FOR STORE-AND-FORWARD COMPUTER-COMMUNICATION NETWORKS*

by Gary L. Fultz and Leonard Kleinrock

Computer Science Department
University of California, Los Angeles, California

ABSTRACT

A study is made of routing techniques applicable to store-and-forward computer networks (e.g., the ARPA Network) in order to show their importance in relation to the theoretical design of these networks and to the performance of existing networks. The major attempt has been to classify routing techniques and to specify their parameters as well as a means of evaluating their performance. Using average message delay as a measure of network performance, a number of routing techniques are compared via theoretical and computer simulation results.

I. INTRODUCTION

This paper considers message flow in a specific class of networks denoted as store-and-forward computer-communication nets. Such nets accept message traffic from external sources (computers) and transmit this traffic over some route within the network to the destination; this transmission takes place over one link at a time, with possible storage of the message at each intermediate switching node due to congestion. One of the fundamental problems in these nets is the routing of messages in an orderly manner to insure their rapid delivery. The requirements for such a system differ considerably from those of the telephone system employing circuit or line switching and from those of military communication networks required to operate in extremely hostile environments.

The study of routing techniques is important because of the central role they play in the design and operation of low cost computer-communication nets. The abstract design of a low cost computer-communication network was first stated by Kleinrock[11] as follows:

minimize T (the average message delay)

$$\text{over the design variables} \begin{cases} \text{link capacity assignment} \\ \text{message priority discipline} \\ \text{routing doctrine} \\ \text{topology} \end{cases} \quad (1)$$

subject to

a suitable cost criterion and external traffic requirement

All of the design variables are interdependent and a general solution technique is unknown, although significant progress has been made for some interesting special cases.[7,11,12,13]

Before the general solution of Eq. (1) can be undertaken, it is important to determine how the variation of the design parameters in this equation influences the average message delay T. Here we address the routing doctrine question. Key areas which require study are: what should a routing technique achieve; how can routing techniques be classified; how are routing algorithms specified; what are the appropriate performance measures and; how are routing algorithms evaluated? Below, we attempt to answer these questions in relation to the selected computer-communication network model.

II. THE COMPUTER-COMMUNICATION NET

In order to properly characterize what an adaptive routing technique (algorithm) should achieve, the universe in which it operates must first be specified. This requires a characterization for computer-communication networks.

The class of networks considered in this paper can be depicted as shown in Figures 1* and 2 and are modeled after the Defense Department's Advanced Research Projects Agency (ARPA) experimental computer network.[7,9,13,16]

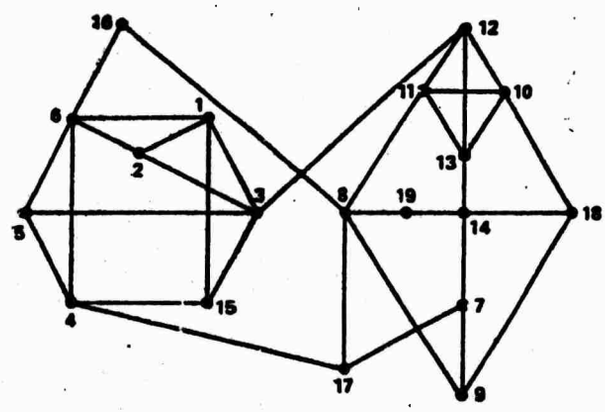


Figure 1. Network Topology

The characteristics of the network model are:

1. Each pair of nodes (N_i, N_j) can be connected by at most one dedicated high-quality (low error rate) full duplex digital communication line.
2. Each communication link has fixed capacity.
3. Each node has finite storage and operates in a store-and-forward fashion.
4. Satellites are not utilized as nodes.[6]

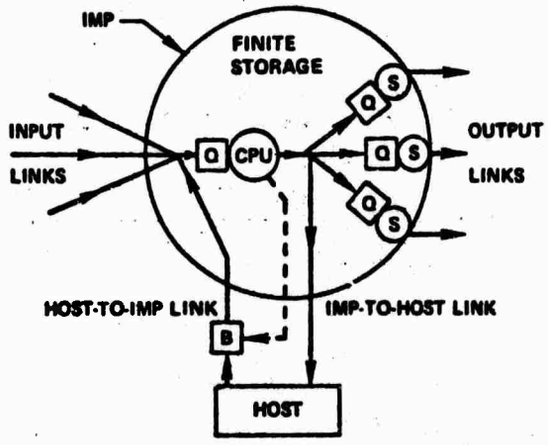
The basic unit of information passed between any pair of nodes is called a "packet" with maximum size of approximately 1000 bits. When a packet is received at a node, it is stored and checked for errors via an error detecting code. If correct and if this node is willing to "accept" the packet, then a positive acknowledgment is sent back to the preceding node indicating this fact; otherwise, a negative acknowledgment is sent back (negative acknowledgments, however, are not used in the ARPA network). When a node receives a positive acknowledgment, it destroys its copy of the packet; otherwise the packet is retransmitted. If a packet is not destined for the node at which it was received, it is relayed (routed) further along its path to a neighboring node.

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285).

*The ARPA network topology has since changed significantly. However, we continue to utilize it in order to compare our current simulation and theoretical results with those contained in Refs. 12 and 13.

The routing procedure determines the path a packet traverses from a source node N_S to a destination node N_D. For example, the paths $\pi_1 = (5,6,16,8)$ and $\pi_2 = (5,4,17,8)$ are two of the many possible paths from $N_S = 5$ to $N_D = 8$ as shown in Figure 1.

The assumed internal structure of a node, shown in Figure 2, consists of a store-and-forward switch referred to as an IMP (Interface Message Processor) and a HOST (external computer system). The function of the IMP is to allocate storage for incoming packets, perform routing for packets which must be relayed, acknowledge accepted packets and perform other routine functions (i.e., packet error checking, circuit fault detection, traffic measurement, etc.). In addition, the CPU routine can block incoming messages from its HOST when sufficient storage is unavailable.



B = SWITCH WHICH CAN BLOCK TRAFFIC FLOW TO THE IMP
CPU = CENTRAL PROCESSING UNIT ROUTINE
Q = QUEUE
S = SERVER (REPRESENTS THE FINITE RATE OF TRANSMISSION ON THE OUTPUT LINKS)

Figure 2. Basic Node Structure

Messages, which originate at a HOST, have a maximum length of approximately 8000 bits. The IMP segments a HOST's message into packets (i.e., as many maximum sized packets as necessary, plus a "remainder" packet). These packets are then handled by the network as independent entities until they reach their destination node. There the packets of a message are collected and the message is reassembled before it is transferred to the destination HOST. Messages which consist of only a single packet are given higher priority than multi-packet messages so that the network can support interactive users.

Using this network model, the message routing requirements for the computer-communication network can be simply stated:

1. Message routing should insure rapid and error-free delivery of messages.
2. The routing technique should adapt to changes in the network topology resulting from node and communication link failures.
3. The routing technique should adapt to varying source-destination traffic loads.
4. Packets should be routed around nodes that are congested or temporarily blocked due to a full storage.

III. CLASSIFICATION OF ROUTING TECHNIQUES

It is desirable to classify network routing techniques in order to gain insight into their structure, complexity and performance; from this, one may then compare them as candidates for operational network algorithms. The two major classifications selected are (1) deterministic, and (2) stochastic techniques. Deterministic routing techniques compute routes based upon a given deterministic decision rule and produce a loop-free routing procedure (i.e., packets cannot become trapped in closed paths). Stochastic techniques, on the other hand, operate as probabilistic decision rules, utilizing topology and either no information about the state of the network (random routing) or estimates of the present state of the network. With these techniques, packets may be trapped in loops for short time periods. Figure 3 shows a more complete classification of the applicable routing techniques.

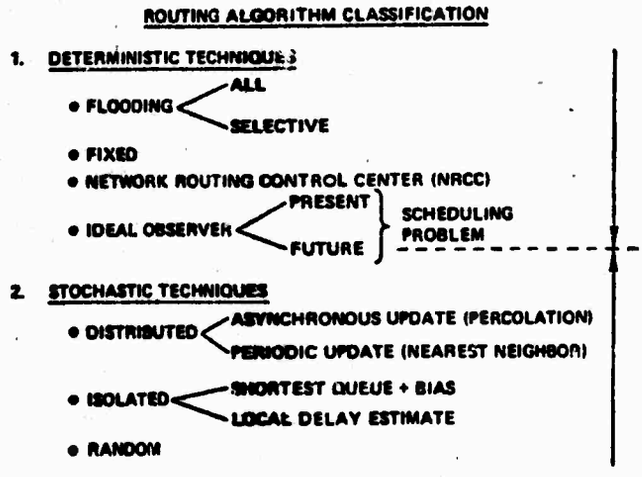


Figure 3. Routing Algorithm Classification

Deterministic Techniques

The four basic deterministic techniques are:

1. Flooding. Each node receiving or originating a packet transmits a copy of it over "all" outgoing links or over a set of "selective" outgoing links; this transmission occurs only after the node has checked to see that it has not previously transmitted the packet, or that it is not the destination of the packet. This technique has been discussed by Boehm and Mobley.[3] Their conclusion is that the inefficiency of this technique is tolerable if one has only a few messages to deliver. However, a large volume of communications traffic necessitates more efficient routing techniques. Another drawback to this technique is that each node requires a mechanism to recognize previously transmitted messages.

2. Fixed Routing. Fixed routing algorithms specify a unique path $\pi(N_S, \ldots, N_D)$ (route) followed by a packet which depends only upon the source-destination node pair (N_S, N_D). To accomplish this, each node has a routing

table similar to that shown in Figure 4. If a packet must be relayed, its destination is used to enter the routing table. The entry contained in the routing table specifies the next unique node in the packet's path. Kleinrock[12,13] and Prosser[15] have examined several of these techniques. Fixed routing techniques require completely reliable nodes and links, except for the occasional retransmission of a packet due to channel bit errors. However, they do allow for highly efficient high volume traffic flow and are very stable.

DESTINATION NODE NUMBER	NEXT NODE NUMBER
1	16
2	11
3	17
4	17
5	17
6	16
7	9
9	9
10	11
11	11
12	11
13	11
14	19
15	17
16	16
17	17
18	9
19	19

Figure 4. Node 6 Routing Table

3. Network Routing Control Center (NRCC). With this technique, one of the network nodes is designated as the NRCC. This center collects performance information about the network operation, computes routing tables and then transmits the appropriate routing table to each node in the network. Computation of the routes by the NCRR is done on a global basis and this insures loop-free paths between all source-destination node pairs. Thus, a fixed routing procedure is maintained between NRCC updates.

There are a number of drawbacks to this technique. By the time the nodes begin using the new routing tables, the performance information that was used in the computation of the routing tables may be out of date in relation to the current state of the network. In addition, transmission costs and vulnerability become significant considerations.

4. Ideal Observer Routing. This technique is essentially a scheduling problem. Each time a new packet enters a node from the HOST, its route is computed to minimize its travel time to its destination node, based upon the complete present information about the packets already in the network and their known routes. If the ideal observer has information about the occurrence of future events, then this information could also be utilized in the computation of the route. This technique is obviously impractical for an operational network, but from a theoretical viewpoint, provides the minimum average message delay to which all other routing techniques may be compared.

Stochastic Techniques

The three basic stochastic techniques are:

1. Random Routing. Random routing procedures are those decision rules in which the choice as to the next node to visit is made according to some probability distribution over the set of neighbor nodes. The set of neighbor nodes utilized in the decision rule can be "all" of the connected nodes or can be based "selectively" over that set of nodes which are in the general direction of the packet's destination.

Kleinrock[11] and Prosser[14] have investigated numerous random routing techniques and have shown that they are highly inefficient in terms of message delay, but are extremely stable (i.e., they are relatively unaffected by small changes in the network structure).

2. Isolated and 3. Distributed Techniques. All of the isolated and distributed routing algorithms operate in basically the same manner. A delay table is formed at each node as shown in Figure 5. The entries $\hat{T}_J(D,L_N)$ are the estimated delays to go from the node under consideration (say node J) to some destination node D using line L_N as the next step in the path to D. A routing table is then formed by choosing, for each row (say the i^{th} row), that output line number $OL_N(i)$ whose value in the delay table is minimum as follows:

$$OL_N(i) = \min_{\{L_N\}} \hat{T}_J(i,L_N) \qquad (2)$$

where $\{L_N\}$ is the set of output line numbers for node J. Figure 5 shows an example.

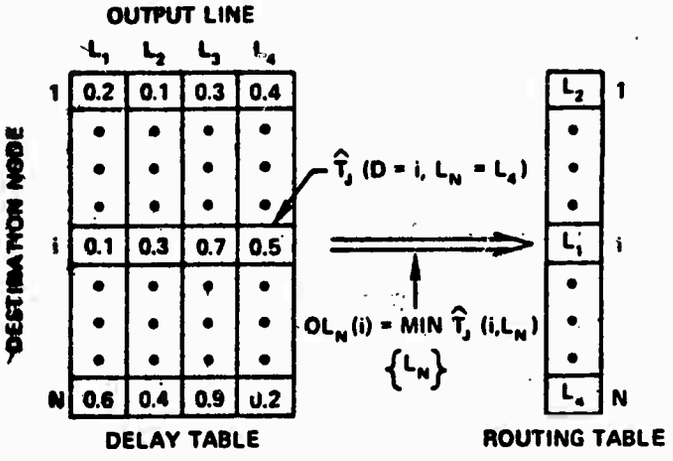


Figure 5. Node J Delay and Routing Tables

The manner in which the estimates $\hat{T}_J(.,.)$ are formed and updated and how often the delay tables are interrogated depends upon the specific structure of the routing algorithm.

In the shortest queue + zero bias algorithm, a packet's route is selected by placing it in the shortest output channel queue. This is essentially Baran's Hot Potato routing concept[1,2]. Since the route selected is independent of the packet's destination, the delay table would require only one row, where the row entries would reflect the output channel queue lengths. The non-zero bias case will be discussed later as a limiting case of a distributed routing technique.

In the local delay estimate algorithm, a packet's route is selected via Eq. (2). The delay table is updated after a packet is received (say at node J) by the following scheme

$$\hat{T}_J(D{=}S,L_N^{(R)})_{new} = K_1 \cdot \hat{T}_J(D{=}S,L_N^{(R)})_{old} + K_2 \cdot TIN(S,J) \qquad (3)$$

where

TIN(S,J) = the Time the packet has spent In the Network traveling from its source node S to the current node J,

$L_N^{(R)}$ = the reverse (outgoing) line corresponding to the forward (incoming) line L_N of the full-duplex pair upon which the packet entered node J,

and

K_1 and K_2 are constants.

This technique, called backwards learning, has been extensively investigated by Baran[1]. Boehm and Mobley[3]

offer modifications to the basic technique (Eq. (3)) to improve its performance.

In the distributed routing techniques classification, all routing algorithms utilize the same basic techniques to compute and update the delay table estimates, but the instants at which these tables are updated and the route selection procedure differs depending upon the particular structure of the algorithm.

There are basically two different mechanisms which cause entries in the delay tables to change: (1) as packets are placed on (or taken off) an output channel queue, all delay table entries in a column corresponding to that output line must be increased (or decreased) to reflect the change in expected delay for the channel; and (2) when delay information from neighboring nodes is utilized to update the delay table estimates. In the latter case, the following procedure is used.

Suppose a decision at node J has been made to inform its neighbors (say N_1, N_2 and N_3, as in Figure 6) of its current minimum estimated delays to reach all nodes within the network. Node J forms a minimum delay vector $\underline{V}_J = (\hat{T}(1), \hat{T}(2), \ldots)$, where the K^{th} component $\hat{T}(K) = \min_{\{L_N\}} \hat{T}_J(K, L_N)$ and transmits $\underline{V}_J$ to its neighbors

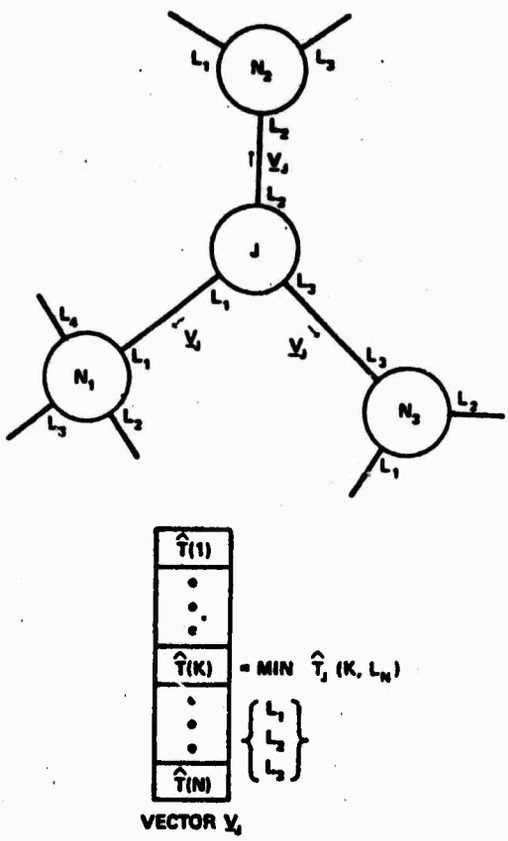


Figure 6. Minimum Delay Vector Transmission

(N_1, N_2 and N_3). Upon receipt of a minimum delay vector, a node (for example N_1) adds its current output line queue length (line L_1 for this example) plus a constant D_p to all entries in the vector $\underline{V}_J$ and replaces column 1 (corresponding to L_1) in its delay table with these new values. Mathematically, the updated delay table entries are

$$\hat{T}_M(D,L_N) = Q(M,L_N) + D_p + \hat{T}(D) \qquad (4)$$

where $Q(M,L_N)$ is the queue length of line number L_N at node M. The constant D_p can be interpreted in two ways. First, if its value equals the average time to transmit a packet over an outgoing channel, then neglecting channel propagation delays, D_p represents the minimum average delay to reach a neighbor node. Secondly, if the delay tables are updated rapidly in a lightly loaded net, then the delay table estimate $\hat{T}_J(D,L_N) = N^*(J,D,L_N)\cdot D_p$ where $N^*(J,D,L_N)$ is the number of lines encountered in the path $\pi(J,D)$ when a packet leaves node J on line L_N. Thus, by varying D_p, we can control the degree of alternate routing and sensitivity of the algorithm to small variations in queue lengths. That is, if D_p is large compared to the average queueing delay in a node, then the path chosen for a message will tend to be one of the paths with smallest $N^*(S,D,\cdot)$.

There are two methods which can be employed to cause the transmission of the delay table update vectors $\underline{V}_J$: (1) The periodic updating algorithm forces these transmissions at a periodic rate R_U (as is currently done in the ARPA network) and (2) the asynchronous updating algorithm allows these transmissions asynchronously; this transmission can occur after the routing of a packet (via Eq. (2)) on line $OL_N(i)$ if $\hat{T}_i(i,OL_N(i))$ has changed by more than a specified amount (a threshold) since the last update occurred. Thus, the delay vectors can percolate throughout the net in a short time period. If the threshold value is excessively large, updating ceases and the asynchronous routing schemes reduce to the shortest queue + bias algorithms (with bias equal to D_p).

The choice of routes is determined as follows: If the update mechanism is periodic, then the set of routes obtained via Eq. (2) is held fixed until the tables are again updated; in the asynchronous case, Eq. (2) is used to determine the route of each packet.

Of all the stochastic techniques, the distributed routing algorithms are the most efficient for handling line and node failures. Once a failure is determined (see Ref. 9 for procedures utilized in the ARPA network), the proper entries in the node delay tables can be forced to remain excessively large as long as the failure persists.

Returning to Figure 3, the arrows on the right-hand side represent (from tail to head) increasing complexity and expected performance of the algorithms. Of all the routing techniques shown, we feel that the distributed stochastic routing techniques have the best potential performance to offer in operational store-and-forward computer-communication networks. These techniques operate essentially as distributed network routing control centers and can adapt rapidly to link and node failures as well as to changing traffic conditions.

IV. NETWORK PERFORMANCE

In order to design optimal computer-communication networks or to assess their performance, one requires quantitative measures of network performance. There are basically two classes of performance measures. The first class does not relate in any simple way to individual messages in the network, but rather to the performance of particular components that compose the network. Examples of such performance measures are: average channel utilization; nodal storage utilization; and channel error rates. Many of these performance measures can be computed analytically. The second class of performance measures relate more directly to individual messages and more definitive statements about overall network perform-

ance can be made. An example of such a performance measure is the measured distribution of time to transmit a message through the net. However, among the possible performance measures, the average message delay is the only one that has yielded to analysis. In addition, it also reflects the following network phenomena in its computation:

- Message delay due to formation of queues within the nodes
- Nodal processing delays
- The decrease in effective channel capacity due to the transmission of acknowledgments and routing information within the network
- Negative acknowledgments causing packet retransmission
- Adaptability of the routing algorithm to varying traffic loads and channel and node failures
- Packet looping caused by momentary errors in estimation of the required routes by the routing algorithm, and
- Nodal storage blockage

In earlier works on communication nets[11] and computer communication networks[12,13], Kleinrock studied such nets using methods from queueing theory[17] which he showed provide an effective method for the computation of the average delay of single packet messages using fixed routing procedures. Fultz[8] has modified these models to more accurately predict the single packet message delays. In addition, he has removed some of the independence assumptions discussed in Ref. 11 in order to handle the multipacket message case. Figures 7 and 8 show a comparison of simulation[10,12] and analytical results[8] for a

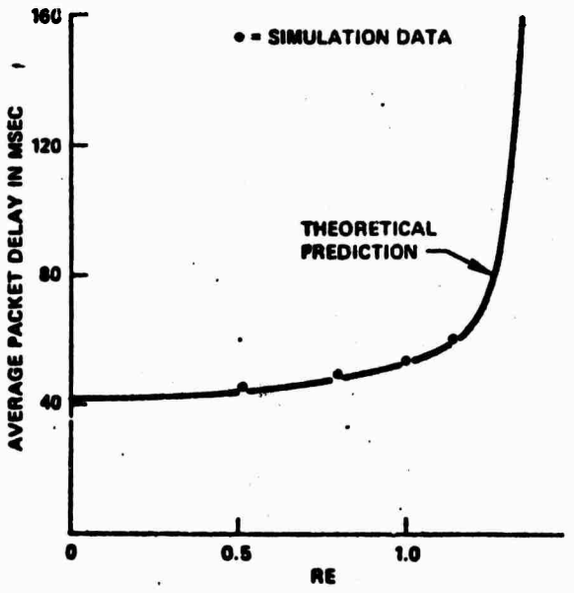


Figure 7. Single-Packet Message Delay

fixed routing procedure utilizing the network configuration shown in Figure 1. Both the analytic and simulation models reflect an assumed traffic matrix [TM] whose entries give the average traffic flow requirements in bits/second between source-destination pairs of nodes. In Figure 7 we have scaled all entries in [TM] by a factor RE (called the Effective Data Rate) which allows us to study the average message delay as a function of network loading. Figure 8 shows the average message delay

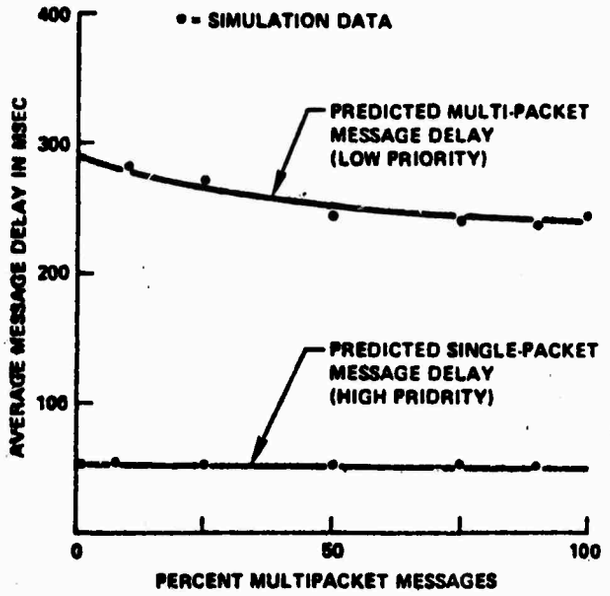


Figure 8. Message Delay Versus Mix (RE = 1)

for the two priority classes as we vary the mix of short (single-packet) high-priority messages and long (multipacket) low-priority messages, while maintaining a constant average input data rate to the entire net. For the fixed routing procedure, we see that the average message delay is adequately predicted by the analytic results. However, when one assesses the performance of stochastic routing techniques, these curves do not indicate typical network performance. Kleinrock[11] and Prosser[14,15] have given methods to analyze random routing procedures. Here we give a method of estimating average single-packet message delay for the isolated and distributed stochastic routing procedures.

We begin by noticing that the isolated and distributed algorithms operate as fixed routing procedures over small periods of time. As time progresses and the algorithm adapts, it utilizes various combinations of fixed routing procedures. Of interest is that fixed routing procedure which minimizes the average message delay for a given network loading factor RE. Figure 9 portrays the average

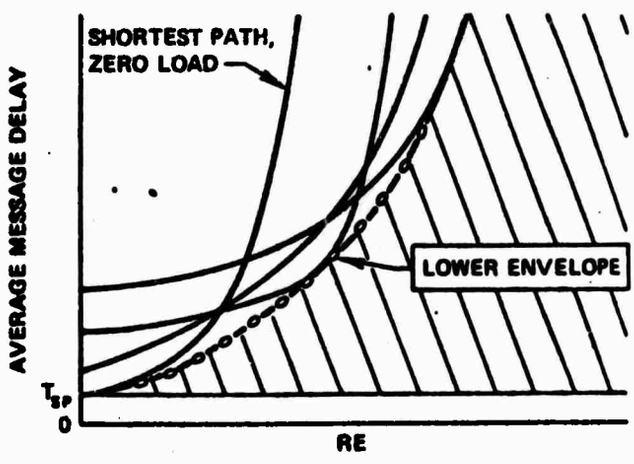


Figure 9. Average Message Delay Profile

single-packet message delay as a function of RE and various fixed routing procedures. The lower envelope of all these delay curves reflects the minimum average message delay utilizing fixed routing algorithms. We have a procedure for computing this lower envelope[8]. The horizontal line of value T_{SP} is the theoretical minimum average message delay and represents a solution of the shortest path problem[5] for RE = 0. The shaded portion of the figure represents a region of operation which can only be penetrated if the stochastic routing algorithm happens to take exquisite advantage of the instantaneous characteristics of message flow within the network to produce a smaller average message delay than the best fixed routing algorithm. To date, none of our simulation results has penetrated this region. This indicates that the lower envelope delay curve is a good measure of attainable performance for stochastic routing algorithms.

For the periodic updating algorithm, there are two parameters which may be adjusted for performance optimization (D_p and the periodic update rate R_U) for any value of RE. Figure 10 shows this performance as a function of D_p for various values of R_U with RE = 1. These delay curves reflect the additional message delay caused by the presence of the routing update traffic flow within the net. For each routing update packet (which contains the vector $\underline{V}_J$), its line transmission time, T_U, utilized in the simulation program was $0.8T_p$, where T_p (= 12.6 msec) is the average line transmission time for a single-packet message (average packet length in bits divided by the line capacity in bits per sec).

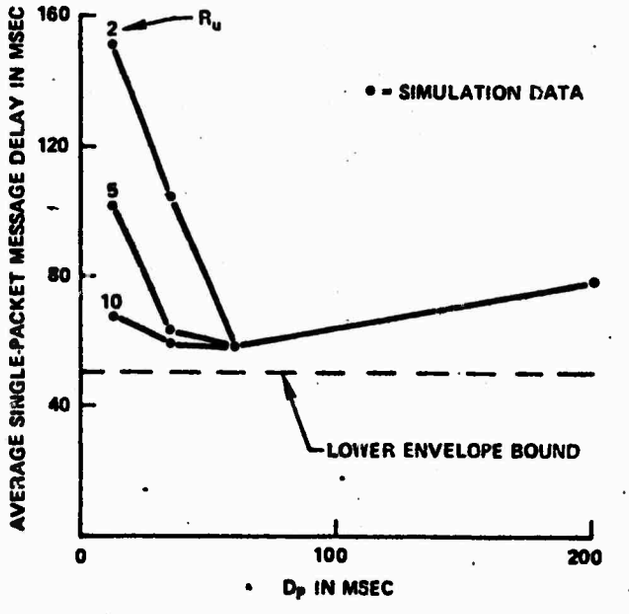


Figure 10. Periodic Updating Algorithm (RE = 1)

For small D_p, the simulation program shows that many loops exist in the fixed routing procedure utilized between delay table updates and thus produces a large average message delay as shown in the figure; the higher values of R_U shown permit better adaptation to the traffic, even offsetting the increase in traffic due to these updates. Although not shown in the figure, limited simulation data indicates that the average delay for R_U = 20 updates/sec is larger than for R_U = 10 updates/sec; thus R_U cannot be increased indefinitely without suffering a loss in performance.

For large D_p (60 msec and greater), little evidence of looping is found. The average message delay for D_p = 200 msec is within two msec of the simulation result at RE = 1 for the fixed routing procedure based upon the solution of the shortest path, zero-load problem. This indicates that the delay table updating cannot, for this value of D_p, adapt to the fluctuations in network traffic so as to lower the average message delay. However, the algorithm can still adapt to line and node failures and the delay and routing tables would reflect these failures. For the simulation data plotted in Figure 10, the minimum average delay occurs at $D_p \approx 60$ msec, which is approximately five times as large as the average line transmission time T_p for a single packet. In the solution of T_{SP} for this network, the longest route also contains five lines. Further investigation is required to determine if there is a similar observable pattern for other values of RE and for variations in the traffic matrix [TM] and network topology.

For the asynchronous updating algorithm, there are also two parameters which can be adjusted for performance optimization (D_p and the threshold values). Here we consider constant thresholds (adaptive thresholds will be considered in the future). The simulated updating procedure operates as follows: A copy of the new minimum delay vector, $\underline{V}_J$, is retained in node J each time it is formed for updating. As packets are routed at node J via Eq. (2), the minimum delay corresponding to $\alpha_{L_N}(i)$ is compared to its corresponding entry $\hat{T}(i)$ in the stored vector $\underline{V}_J$ as shown below.

$$|\hat{T}_J(i) - \hat{T}_J(i,\alpha_{L_N}(i))| = \Delta\hat{T}_J(i) \qquad (5)$$

If $\Delta\hat{T}_J(i) \geq$ threshold, then the update procedure is invoked as shown in Figure 6. Otherwise, no update occurs. The motivation, of course, for utilizing thresholds is to sense changes in the traffic distribution (delay) and only update when these changes are pertinent as opposed to the periodic updating algorithm which forces updates even when the delay tables remain static. Figure 11 shows the algorithm performance as a function of D_p

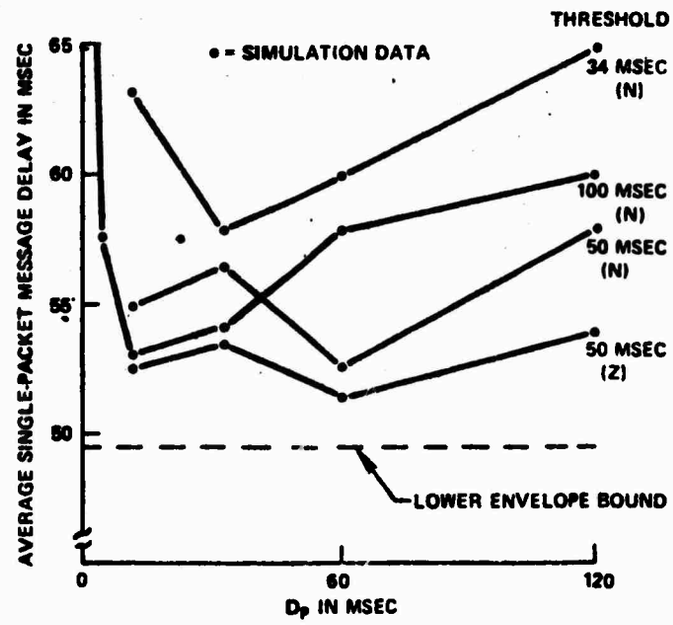


Figure 11. Asynchronous Updating Algorithm (RE = 1)

for various threshold values with RE = 1. The curves labeled (N) indicate normal operation of the algorithm ($T_U = 0.8T_p$), while the curve labeled (Z) corresponds to $T_U = 0$. Thus, the difference between the two 50 msec threshold curves represents the increase in average message delay due to the presence of the update traffic within the net.

The asynchronous update algorithm does not exhibit the distinct minimum message delay as a function of D_p as found for the periodic update algorithm. Also, no correlation was found between the number of updates and message delay for a fixed threshold value, even though the number of updates increased as D_p increased (except for the dip in the 50 msec threshold curves at D_p = 60 msec). For a fixed $D_p \geq 60$ msec, there is a correlation between average message delay and threshold, the minimum being at approximately the 50 msec threshold, which lies between the 34 and 100 msec thresholds.

Perhaps the most interesting delay curve shown in Figure 11 is that for a threshold of 100 msec. For $D_p \leq 34$ msec, no updates occurred during the simulation; thus the algorithm operation reduced to the shortest queue + bias class. However, line and node failures would cause the algorithm to update. It is quite possible that the threshold test (Eq. (5)) could be eliminated and updating forced only when a line or node failure is recognized. This requires further investigation. For $D_p \leq T_p$ msec, the algorithm becomes highly unstable and many loops appear in the routing. This accounts for the large increase in delay, as the figure indicates.

Finally, Figure 12 shows the best simulated performance of three routing algorithms (periodic updating, asynchronous updating and shortest queue + bias) as a function of the network traffic loading factor RE. The lower envelope represents the achievable average message delay for the best fixed routing scheme (although it would vary with RE), but has not as yet been simulated. However, the results shown in Figure 7 indicate that simulation should agree with this theoretical curve.

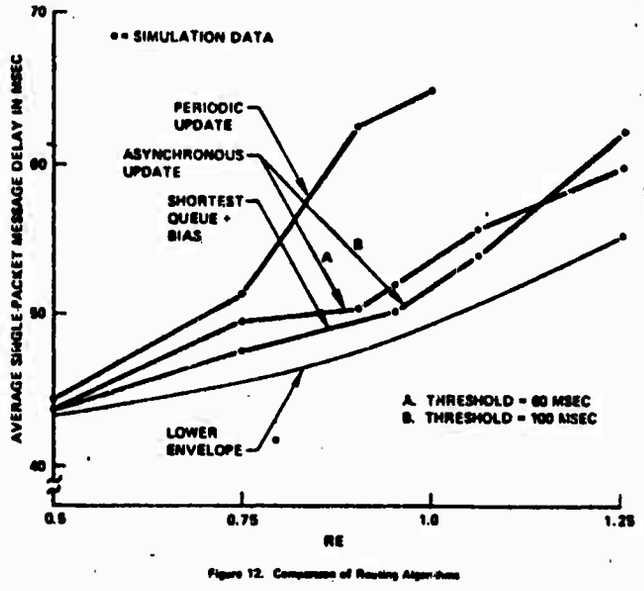


Figure 12. Comparison of Routing Algorithms

The periodic updating algorithm is shown to be inferior in performance to the other algorithms simulated. Apparently, utilizing a fixed routing procedure between updates causes increased congestion within the network and thus increases message delay.

For a moderate threshold value (100 msec), the asynchronous updating algorithm achieves the same performance as the shortest queue + bias algorithm (because no delay table updates were initiated during the simulation). The 60 msec threshold value produces a very interesting result. As RE approaches 1.25, the asynchronous updating algorithm performs better than the shortest queue + bias algorithm. This shows that the algorithm is utilizing the information contained in the minimum delay vectors V_J to adapt to the fluctuations in network traffic flow. Further, it indicates that the presence of the delay table updating traffic within the net does not necessarily cause an increase in the average message delay.

Before a more detailed comparison can be made among the algorithms, further understanding of the relationship between D_p and R_U or the threshold value must be gained. In addition, line and node failures must be simulated in order to determine how rapidly the algorithms adapt and what average message delay they produce.

V. CONCLUSIONS

We have presented a meaningful overview of routing techniques available for computer-communication networks and have developed the structure of routing algorithms which appear to be the most promising for operational networks. The main thrust of our research has been to develop models of network performance and routing algorithms and compare their performance via computer simulation. Moreover, preliminary measurement data (time delay measurements, degree of alternate routing, etc.), collected by Cole[4] on the ARPA network, indicates general agreement with our simulation results. We are now in a position to compare our analytic and simulation models with real network performance data.

We have demonstrated that fixed routing procedures perform most effectively from among our many comparisons; however, such procedures cannot adapt to variations in network traffic and topology. The adaptability of our distributed stochastic algorithms provides efficient performance under such variations and appears as strong candidates for use in store-and-forward computer-communication nets.

REFERENCES

1. Baran, P., "On Distributed Communications," The Rand Corporation, Series of 11 Memoranda, August 1964.

2. Boehm, S., and P. Baran, "Digital Simulation of Hot-Potato Routing in a Broadband Distributed Communication Network," The Rand Corporation, Memorandum, Rm-3103-PR, August 1964.

3. Boehm, B. W., and R. L. Mobley, "Adaptive Routing Techniques for Distributed Communications," The Rand Corporation, Memorandum, Rm-4781-PR, 1966.

4. Cole, G., "Computer Network Measurements: Theory and Design," Ph.D. Dissertation, Computer Science Department, University of California at Los Angeles, to be published, 1971.

5. Dijkstra, E. W., "A Note on Two Problems in Connection with Graphs," Numerische Mathematik, pp. 269-271, Vol. 1, 1959.

6. Ferrell, C. W., and P. J. Knobe, "The Impact of Satellite Communications on Computer Networks," Proceedings of the Computers and Communications Conference, Rome, New York, September 30-October 2, 1969.

7. Frank, H., I. T. Frisch and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," AFIPS Conference Proceedings, Spring Joint Computer Conference, May 1970.

8. Fultz, G. L., "Adaptive Routing Techniques for Store-and-Forward Message-Switching Computer-Communication Networks," Ph.D. Dissertation, Computer Science Department, University of California at Los Angeles, to be published, 1971.

9. Heart, F. E., R. E. Kahn, S. M. Ornstein, W. R. Crowther and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, Spring Joint Computer Conference, May 1970.

10. IBM Corporation, "General Purpose Systems Simulator III, User's Manual," Form H20-0163.

11. Kleinrock, L., Communication Nets; Stochastic Message Flow and Delay, New York: McGraw-Hill, 1964.

12. Kleinrock, L., "Models for Computer Networks," Proceedings of the International Communications Conference, Boulder, Colorado, pp. 21-9 to 21-16, June 1969.

13. Kleinrock, L., "Analytic and Simulation Methods in Computer Network Design," AFIPS Conference Proceedings, pp. 569-579, Spring Joint Computer Conference, May 1970.

14. Prosser, R. J., "Routing Procedures in Communication Networks, Part I: Random Procedures," IRE Transactions on Communication Systems, CS-10, pp. 322-329, 1962.

15. Prosser, R. J., "Routing Procedures in Communication Networks, Part II: Director Procedures," IRE Transactions on Communications Systems, CS-10, pp. 329-335, 1962.

16. Roberts, L. G., and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, pp. 543-549, Spring Joint Computer Conference, May 1970.

17. Saaty, T. L., Elements of Queueing Theory with Applications, New York: McGraw-Hill, 1961.

APPENDIX E

NODAL BLOCKING IN LARGE NETWORKS

by J. F. Zeigler and L. Kleinrock

NODAL BLOCKING IN LARGE NETWORKS*

by Jack F. Zeigler and Leonard Kleinrock

Computer Science Department
University of California, Los Angeles, California

ABSTRACT

A theoretical study is given for store-and-forward communication networks in which the nodes have finite storage capacity for messages. A node is "blocked" when its storage is filled, otherwise it is "free." A two-state Markov model is proposed for each node, and the number of blocked nodes in the network is shown also to have a two-state Markov process representation. Digital computer simulations substantiate the theoretical results.

INTRODUCTION

Consider a store-and-forward communication network (e.g., see Refs. 1-5) consisting of nodes having finite storage space for messages. During periods of high traffic intensity this storage can be expected to fill from time to time. In this condition the node must refuse incoming messages (which might be accomplished by sending negative acknowledgments) and we then say that the node is "blocked."

As soon as one message is transmitted by a blocked node, it becomes a "free" node. It remains in this state as long as there is at least one empty space in storage that could be used by an arriving message. When the storage fills again, the node re-enters the blocked state.

THE MODEL

Figure 1 shows a simplified model of such a node in the terminology of the ARPA network[1-5]. The Interface Message Processor (IMP), when free, accepts messages into its main storage from two sources: (1) other IMPs like itself, and (2) a HOST which generates and receives messages (as a source and terminal) and communicates with the rest of the network by means of the IMP. Message bits are sent in parallel to the message buffer serving the appropriate output line, as determined by the final destination of the message, and are then transmitted serially to that neighbor. Any of these output lines can become blocked, thus preventing their use.

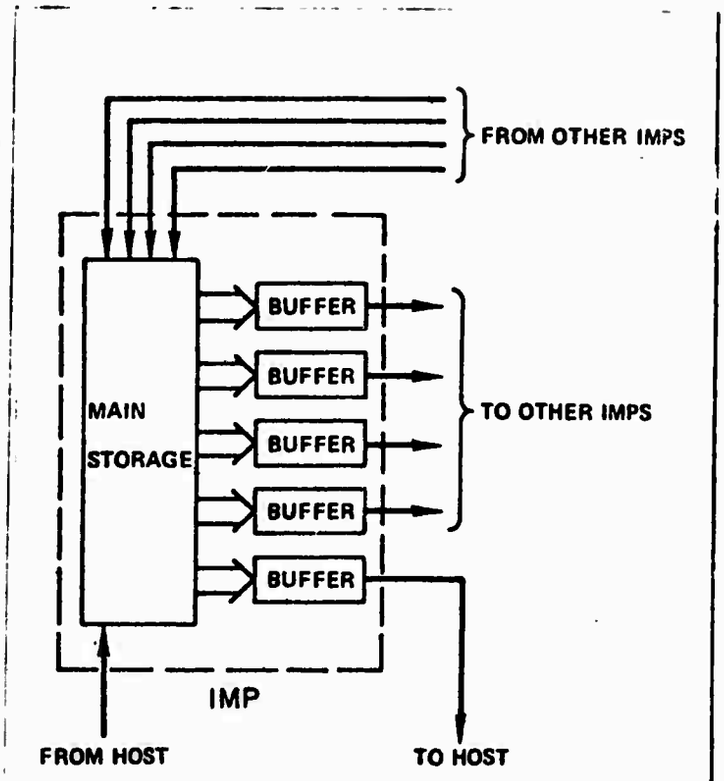


Figure 1. Schematic of a Node

In this paper we study nodal blocking caused by the finite storage room for messages in the IMP and the overutilization of the system. By overutilization, we mean that when the node is accepting messages, its average arrival rate equals or exceeds its average service rate (which is the total output channel capacity divided by the average message length). Elementary queueing theory[6] shows that if (1) the system is underutilized, and (2) there is storage space for approximately twenty messages or more, then under fairly general conditions there will be essentially no blocking.

Nodal blocking is a transient effect which should occur only at peak hours during the day in a well-designed system, but once started it could propagate in both space and time. The analysis of this propagation is difficult for at least three reasons. First, it involves networks of queues, for which only stationary results at best can generally be obtained. Second, the pertinent stochastic processes are dependent, for if a node becomes blocked, it cannot accept messages from its neighbors and their storage will tend to fill at a faster rate. Finally, it is a transient queueing problem and even the simplest of these is very difficult to solve. (For example, the queueing system with Markovian arrivals, a single exponential server, and unlimited waiting room has modified Bessel functions in its time dependent solution[6].)

Since we cannot solve the problem exactly, our goal is to make good approximations that allow us to analyze the system and characterize its blocking behavior in some way. To this end we make the following assumptions:

1. The HOST cannot become blocked (it is an infinite sink)

2.a. Input traffic from the HOST is Poisson

b. Traffic on all lines has the same average rate so that total average traffic into each node is σ messages/sec.

3.a. Message lengths are exponentially distributed

*This work was supported in part by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285) and a National Science Foundation Traineeship.

b. Service (transmission) time on any line is therefore exponentially distributed such that for a node with k blocked neighbors, the rate at which messages exit from that node is $\mu^{(k)}$ messages/sec.

4. Probability of an empty queue in the IMP is approximately zero (since the system is assumed to be overutilized)

ANALYSIS AND RESULTS

Under these assumptions we arrive at a simplified blocking model for a node in the network, and its description as a two-state Markov process is given in Figure 2. If the node is blocked, i.e., in state b,

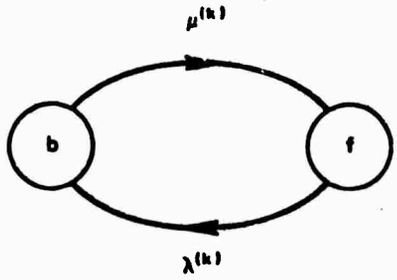


Figure 2. Blocking Model for an Imp

it becomes free in the next instant of time Δt with probability $\mu^{(k)}\Delta t$ where k is the number of blocked neighbors it is experiencing at that time. Similarly, if the node is free, i.e., in state f, it becomes blocked in the next instant of time Δt with probability $\lambda^{(k)}\Delta t$ where k is again the number of blocked neighbors.

Below we show the appropriateness of this model. First, we require the Laplace transform of the interdeparture time probability density $\equiv D(s)$. For any node let $\rho \equiv P_r$[non-empty node] and let the Laplace transform of the probability density of the interarrival time process be $A(s)$. Because we have assumed that the service time is exponential with parameter $\mu^{(k)}$, we know that the Laplace transform of the departure process, conditioned on a non-empty system is $\mu^{(k)}/(s + \mu^{(k)})$. Therefore,

$$D(s) = \frac{\rho\mu^{(k)}}{s + \mu^{(k)}} + (1 - \rho)A(s)\frac{\mu^{(k)}}{s + \mu^{(k)}} \qquad (1)$$

By assumption (4) we have $\rho \approx 1$

$$\therefore D(s) \approx \frac{\mu^{(k)}}{s + \mu^{(k)}} \qquad (2)$$

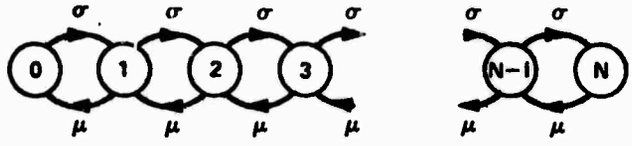
which says that the departure process is a Poisson stream.

We have assumed that the traffic on all lines has the same average rate. If, for example, every node has exactly four neighbors and one HOST, then there are five output lines from each node. All of these lines are equivalent (except that the HOST cannot become blocked) and, by the assumption of exponential message lengths, the departure process from each output line constitutes a Poisson stream when that neighbor is not blocked.

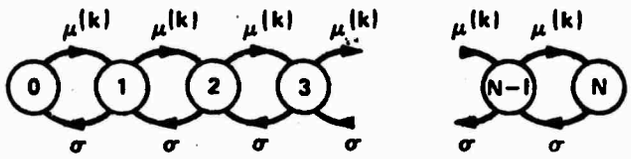
$$\therefore \mu^{(k)} = \frac{5 - k}{5}\mu^{(0)} \qquad k = 0,1, \ldots, 4 \qquad (3)$$

where $\mu^{(0)}$ is a given system parameter and represents the maximum message departure rate from a node. This set of numbers is merely an illustration; any combination can be treated by this model. These results show that we can approximate the time spent in the blocked state as being exponentially distributed with parameter $\mu^{(k)}$.

The time spent in the free state, however, is distributed as the busy period in a queueing system with finite queueing room for customers, as we now show. Consider the state transition diagram or Markov chain model for our single node queueing system shown in Figure 3a.



a) QUEUE STATE TRANSITIONS



b) DUAL QUEUE STATE TRANSITIONS

Figure 3

The numbers inside the circles represent the number of customers in the node. Customers arrive in a Poisson fashion with parameter σ, and depart after receiving service (exponentially distributed with an average of $1/\mu$ seconds). A busy period begins when a customer arrives to find an empty system (at which time he immediately enters the service facility). Customers arriving during his service time form a queue behind him. With each arrival the system moves to the right along the state transition diagram, because the number in the system is increased by one, and with each service completion, i.e., departure, it moves to the left. The busy period ends the first time the system goes empty after initiation of the busy period.

We now consider a dual queue in which the roles of service and arrival are reversed, and the numbers inside the circles now represent the number of empty places in storage that could be used by arriving messages (see Figure 3b). The free period of the IMP begins with the

departure of a message from a previously filled system, i.e., no empty places for arriving messages. With the transmission (departure) the system moves from state 0 to state 1. It continues to move to the right with each transmission and to the left with each arrival. The free period ends the first time the system returns to the 0 state. The correspondence between the primal and dual queues is perfect; thus any results obtained for the busy period in the primal system are applicable to the dual queue free period in the IMP simply by substituting $\mu^{(k)}$ for σ and σ for μ.

The busy period for a finite queueing room system is difficult to obtain, but the result for unlimited queueing room is well known. The probability density of the length t of the busy period in such a system is

$$p(t) = \frac{1}{t\sqrt{\rho}} e^{-(\sigma+\mu)t} I_1(2t\sqrt{\sigma\mu}) \qquad (4)$$

where ρ, the utilization factor $= (\sigma/\mu) < 1$ and $I_1(x)$ is the modified Bessel function of the first kind, of order one[7]. If the size of the queueing room is greater than 20, the solution for unlimited queueing room is a good approximate solution to the limited queueing room problem. (This follows since we have assumed P[empty IMP] ≈ 0. But the P[empty IMP] corresponds to the probability of being in state N(i.e., all N spaces are empty) in Figure 3b, and thus an increase in N will not seriously affect our results.) Since we have assumed overutilization, we have $(\mu^{(0)}/\sigma) < 1$, and we are justified in substituting this (or $\mu^{(k)}/\sigma$) for ρ. Thus we get the following for the probability density of the length t of the time spent in the free state:

$$p(t) = \frac{1}{t}\sqrt{\frac{\sigma}{\mu^{(k)}}} e^{-(\sigma+\mu^{(k)})t} I_1(2t\sqrt{\sigma\mu^{(k)}}) \qquad (5)$$

As the ratio $\mu^{(k)}/\sigma$ approaches 0, i.e., as the system becomes more overutilized, this density approaches that of the exponential distribution. To arrive at a more tractable model, we approximate the free period distribution by the exponential distribution having the same mean value. The mean value of the busy period in the original system is easy to obtain, and is given by $1/\mu(1-\rho)$. Therefore, as an approximation to the free period in the IMP, we take an exponential distribution with mean value $1/(\sigma-\mu^{(k)})$, i.e., with a parameter.

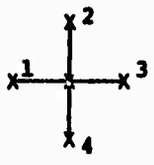
$$\lambda^{(k)} = \sigma - \mu^{(k)} \qquad (6)$$

For the marginal case, $\sigma = \mu^{(0)}$, elementary queueing theory[6] shows that we must take

$$\lambda^{(0)} = \frac{\sigma}{N} \qquad \text{for } \sigma = \mu^{(0)} \qquad (7)$$

where N is the size of the storage capacity in the IMP.

Our model for the blocking IMP is thus a two-state Markov process or, in the language of renewal theory, an alternating Poisson renewal process[8]. One way to describe the dynamics of a network of such nodes is to examine the probability that any given node is blocked at some time t. Consider a node with four neighbors numbered 1 to 4:



Let $P^k(t) = P[k \text{ neighbors blocked at time } t]$ (8)

and let $p(t) = P[\text{node blocked at time } t]$ (9)

Then, from elementary considerations, we have (correct to within $0(\Delta t)$)

$$p(t+\Delta t) = (1-p(t)) \sum_{k=0}^{4} P^k(t)\lambda^{(k)}\Delta t + p(t)\left(1-\sum_{k=0}^{4} P^k(t)\mu^{(k)}\Delta t\right)$$

where from Eq. (3) $\mu^{(k)} = \mu^{(0)} - (k/5)\mu^{(0)}$

and from Eq. (6) $\lambda^{(k)} = \sigma - \mu^{(k)} = \sigma - \mu^{(0)} + (k/5)\mu^{(0)}$

for $\sigma > \mu^{(0)}$.

We also note that

$$\lambda^{(k)} + \mu^{(k)} = \sigma \qquad (10)$$

Thus, $$\frac{p(t+\Delta t) - p(t)}{\Delta t} = (1 - p(t)) \sum_{k=0}^{4} P^k(t)\lambda^{(k)} - p(t) \sum_{k=0}^{4} P^k(t)\mu^{(k)}$$

Letting Δt approach 0, we have

$$\frac{dp(t)}{dt} = -p(t) \sum_{k=0}^{4} P^k(t)(\lambda^{(k)} + \mu^{(k)}) + \sum_{k=0}^{4} P^k(t)\lambda^{(k)}$$

$$= -\sigma p(t) \sum_{k=0}^{4} P^k(t) + \sum_{k=0}^{4} P^k(t)\left(\sigma - \mu^{(0)} + \frac{k}{5}\mu^{(0)}\right)$$

$$= -\sigma p(t) + \sigma - \mu^{(0)} + \frac{\mu^{(0)}}{5} \sum_{k=0}^{4} kP^k(t) \qquad (11)$$

This can be simplified by noting that

$$E[\text{number of blocked neighbors at time } t] = \sum_{k=0}^{4} kP^k(t) \qquad (12)$$

where E denotes expectation.

Define the indicator function

$$fn(t) = \begin{cases} 1 & \text{if node n is blocked at time t} \\ 0 & \text{otherwise} \end{cases}$$

Now let

$$pn(t) = P[\text{node n is blocked at time t}]$$

then $$E[fn(t)] = pn(t) \qquad (13)$$

Further, from Eq. (12), we have that

$$\sum_{k=0}^{4} kP^k(t) = E\left(\sum_{n\in N} fn(t)\right)$$

$$= \sum_{n\in N} E(fn(t)) \qquad (14)$$

where N is the set of neighbors for this node (of which there are four). From Eqs. (13) and (14) we get

$$\sum_{k=0}^{4} kP^{k}(t) = p_1(t) + p_2(t) + p_3(t) + p_4(t) \qquad (15)$$

Finally, from Eqs. (11) and (15) we have the result

$$\frac{dp(t)}{dt} = -\sigma p(t) + \sigma - \mu^{(0)} + \frac{\mu^{(0)}}{5}(p_1(t) + p_2(t) + p_3(t) + p_4(t)) \qquad (16)$$

This relation can also be derived from epidemiology by considering nodal blocking as a deterministic epidemic without migration and with but two kinds of individuals, infected and susceptible[9].

Adjacent nodes have nearly equal probabilities of being blocked. Consider the case when all of these probabilities are exactly equal (as an approximation). Then from Eq. (16)

$$\frac{dp(t)}{dt} = -\sigma p(t) + \sigma - \mu^{(0)} + \frac{4}{5}\mu^{(0)}p(t)$$

$$= -(\sigma - \tfrac{4}{5}\mu^{(0)})p(t) + \sigma - \mu^{(0)}$$

which has the solution

$$p(t) = \left[p(0) - \frac{\sigma - \mu^{(0)}}{\sigma - \frac{4}{5}\mu^{(0)}}\right] e^{-(\sigma - \frac{4}{5}\mu^{(0)})t} + \frac{\sigma - \mu^{(0)}}{\sigma - \frac{4}{5}\mu^{(0)}} \qquad (17)$$

Now consider the alternating Poisson renewal process shown in Figure 4. There are two states, called (B) and free (F). If the system is in the blocked state at time t, it goes free in the next instant of time Δt with probability $(\mu^{(0)}/5)\Delta t$. In similar fashion, the probability that it leaves the free state and re-enters the blocked state is $(\sigma - \mu^{(0)})\Delta t$. Therefore, the probability that it is in the blocked state at time $t + \Delta t$ is

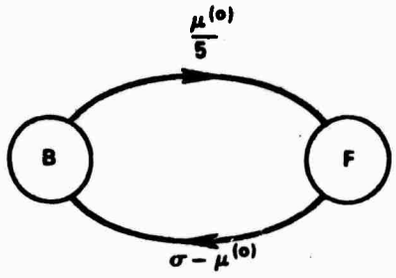


Figure 4. Network Model

$$p_B(t + \Delta t) = p_B(t)(1 - \frac{\mu^{(0)}}{5}\Delta t) + (1 - p_B(t))(\sigma - \mu^{(0)})\Delta t$$

$$\therefore \frac{dp_B(t)}{dt} = -p_B(t)(\sigma - \tfrac{4}{5}\mu^{(0)}) + (\sigma - \mu^{(0)})$$

or

$$p_B(t) = \left[p_B(0) - \frac{\sigma - \mu^{(0)}}{\sigma - \frac{4}{5}\mu^{(0)}}\right] e^{-(\sigma - \frac{4}{5}\mu^{(0)})t} + \frac{\sigma - \mu^{(0)}}{\sigma - \frac{4}{5}\mu^{(0)}} \qquad (18)$$

This is the same as Eq. (17) which was obtained for the probability that a node is blocked at time t! In a large homogeneous system the fraction of blocked nodes may be closely approximated by the probability that any one of them is blocked. Therefore, the fraction of blocked nodes at time t in a large uniformly connected (i.e., two-dimensional lattice) network is approximately equal to the probability that the two-state Markov process shown in Figure 4 is in the blocked state at time t. Thus we may take this two-state Markov process as a model for the network.

So far we have presented only aggregate results. To obtain the probability that any given node in the network is blocked at time t we must consider a system of equations of the form

$$\frac{dp_i(t)}{dt} = -\sigma p_i(t) + \sigma - \mu^{(0)} + \frac{\mu^{(0)}}{5}(p_j(t) + p_k(t) + p_\ell(t) + p_m(t))$$

for each node i in the network with neighbors j, k, ℓ, and m. These equations are obviously of the form

$$\dot{P}(t) = AP(t) + C \qquad (19)$$

If there are n nodes in the net, then P(t) is the n x 1 matrix whose i^{th} component is the probability that node i is blocked at time t. A is an n x n constant matrix and C is an n x 1 constant matrix. The solution is well known:

$$P(t) = e^{At}P(0) + A^{-1}(e^{At} - I)C \qquad (20)$$

For a small net this solution poses no difficulty, but for a large one the required matrix computations rapidly get out of hand. There are some special cases which are solvable, however, and we obtain the solution for one of these below.

Consider a network consisting of 1024 nodes arranged in a 32 x 32 grid. For this system the matrix A is 1024 x 1024 and takes the following form:

$$A = \begin{bmatrix} D & A & & & \\ A & D & A & & \\ & A & D & A & \\ & & \ddots & \ddots & \ddots \\ & & & A & D & A \\ & & & & A & D \end{bmatrix} \qquad (21)$$

$$\text{where } D = \begin{bmatrix} a & b & & & \\ b & a & b & & \\ & b & a & b & \\ & & \ddots & & \\ & & & b & a \end{bmatrix}_{n \times n} \qquad (22)$$

and $$\Lambda = bI_n \qquad (23)$$

where $a = -\sigma$, $b = \frac{\mu^{(0)}}{5}$, and I_n is the

$n \times n$ identity matrix (25)

This observation holds for a square grid with any number of nodes n on a side. (See the Appendix for the complete solution for P(t) for arbitrary n.) This case of n = 1024 was simulated and is described in the following section.

SIMULATION RESULTS

Simulation of a network of 1024 nodes employing the Markovian inter-event time assumption substantiates the approximations described in the theoretical results above. Two different simulation programs have been run on the UCLA XDS Sigma-7 computer.* The first was for a network arranged in a square grid 32 x 32. Each node is connected to its four nearest neighbors (a lattice) except in the case of nodes along the border which have only three nearest neighbors (or two nearest neighbors in the case of the four corner nodes). When a node changes state, new event times are chosen for it and for all of its nearest neighbors based on the new number of blocked neighbors. The memoryless property of the exponential distribution simplifies the calculations.

The second program simulated a randomly connected graph in which each node was given exactly four neighbors.

Comparison of the two-state Markov process model and the simulation results for the lattice and the random graph are shown in Figure 5 for one set of parameters σ and $\mu^{(0)}$ starting from a net that is completely blocked. Figure 6 shows the results when the network begins with all of its nodes in the free state. In Figure 7 results are compared for the model and the two-dimensional integer lattice in which each node is assumed to have eight neighbors. This was accomplished by extending the nearest neighbor definition to include nodes which are diagonally adjacent. The results in Eq. (18) are extended in the obvious way. Figure 8 compares simulation results on the lattice of degree four, when a free node with k blocked neighbors is considered k-fourths blocked, to the predicted trajectory based on a non-linear "partial blocking" model. The agreement with the simulations is generally good, and the model is sufficiently general to treat a variety of cases.

CONCLUSIONS

Two new models that may have application to store-and-forward communication networks are presented in this paper. The probabilistic model for nodal blocking due to finite storage space is shown in Figure 2. The second model, and the main result of this work, is that the fraction of blocked nodes in a network of such nodes has

*During simulation the net activity was displayed on a Digital Equipment Corporation 340 Precision Display CRT.

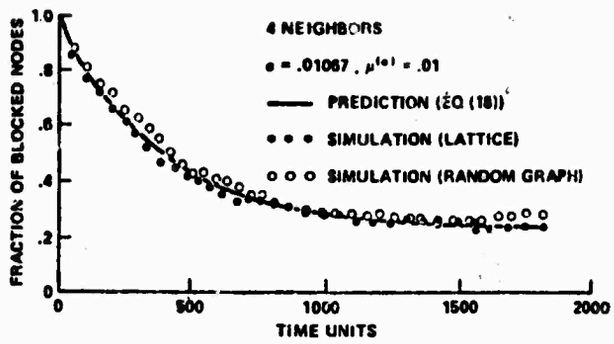


Figure 5. Fraction of Blocked Nodes I

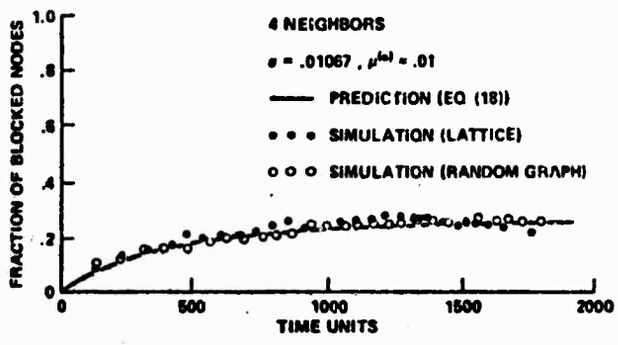


Figure 6. Fraction of Blocked Nodes II

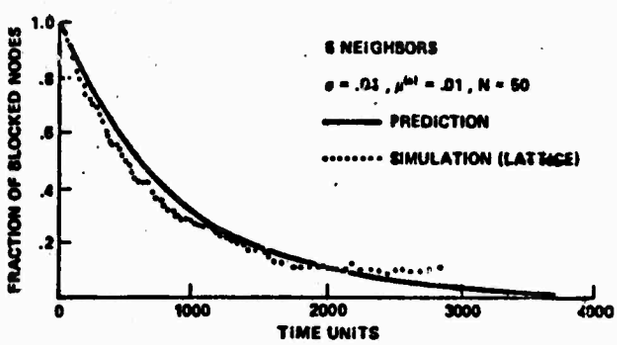


Figure 7. Fraction of Blocked Nodes III

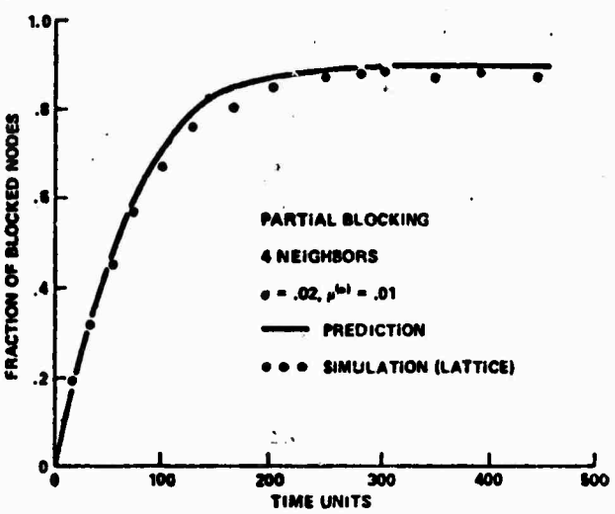


Figure 8. Fraction of Blocked Nodes IV

Eq. (18)). Figures 5-8 verify that the network model compares well with results obtained from the simulation of a network of two-state Markovian nodes in which the time spent in either state is a function only of the state and the number of blocked neighbors. Finally, the model is sufficiently general to treat a variety of network configurations and parameters.

ACKNOWLEDGMENTS

The authors would like to thank Professor David Cantor and Mr. Frank Kontrovich for their frequent help in the analysis of this problem.

REFERENCES

1. L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," AFIPS Conference Proc., vol. 36, pp. 543-549, May 1970.
2. F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther and D. C. Walden, "The interface message processor for the ARPA computer network," AFIPS Conference Proc., vol. 36, pp. 551-567, May 1970.
3. L. Kleinrock, "Analytic and simulation methods in computer network design," AFIPS Conference Proc., vol. 36, pp. 569-578, May 1970.
4. H. Frank, I. T. Frisch and W. Chou, "Topological considerations in the design of the ARPA computer network," AFIPS Conference Proc., vol. 36, pp. 581-587, May 1970.
5. C. S. Carr, S. D. Crocker and V. G. Cerf, "HOST-HOST communication protocol in the ARPA network," AFIPS Conference Proc., vol. 36, pp. 589-597, May 1970.
6. D. R. Cox and W. L. Smith, Queues, London: Methuen, 1968.
7. F. B. Hildebrand, Advanced Calculus for Applications, Englewood Cliffs, N.J.: Prentice-Hall, 1965, p. 150.
8. D. R. Cox, Renewal Theory, London: Methuen, 1962.
9. M. S. Bartlett, Stochastic Population Models in Ecology and Epidemiology, London: Methuen, 1960.
10. U. Grenander and G. Szego, Toeplitz Forms and Their Applications, Los Angeles: Univ. of California Press, 1958.

APPENDIX

We must first find the eigenvalues γ_v of D which are the solutions of $|D - \gamma I| = 0$. Let a stand for $a - \gamma$ in D. we wish to find the zeros of the determinant of D. Expanding by the elements of the top row, we note the following recurrence relation for the determinant Δ_n of the n x n matrix D:

$$\Delta_n = a\Delta_{n-1} - b^2\Delta_{n-2}$$

with initial conditions $\Delta_1 = a$, $\Delta_0 = 1$, $\Delta_{-1} = 0$. Following Grenander and Szego[10] we substitute $a = 2b \cos\theta$, assume a solution of the form $\Delta_n = \rho^n$, and solve the resulting quadratic in ρ. After satisfying the initial conditions the result is simply

$$\Delta_n = b^n \frac{\sin(n+1)\theta}{\sin\theta}$$

which vanishes for $\theta = v\pi/n+1 \qquad v = 1,2,..,n$

Therefore, the eigenvalues of D are

$$a - 2b\cos\frac{v\pi}{n+1} \qquad v = 1,2,..,n$$

which are all distinct. The eigenvectors are the solutions of

$$\begin{bmatrix} a & b & & & \\ b & a & b & & \\ & b & a & b & \\ & & \cdots & & \\ & & & b & a \end{bmatrix} \begin{bmatrix} X_{v1} \\ X_{v2} \\ X_{v3} \\ \cdots \\ X_{vn} \end{bmatrix} = \gamma_v \begin{bmatrix} X_{v1} \\ X_{v2} \\ X_{v3} \\ \cdots \\ X_{vn} \end{bmatrix}$$

It is easy to verify that the normalized solutions are

$$X_{vk} = \frac{(-1)^{n-k}}{\sqrt{\frac{n+1}{2}}} \sin\frac{kv\pi}{n+1}$$

so that the (i,j) element of e^D

$$e^D_{i,j} = \sum_{v=1}^{n} e^{\gamma_v} X_{vi} X_{vj}$$

and $$D^{-1}_{i,j} = \sum_{v=1}^{n} (\gamma_v)^{-1} X_{vi} X_{vj}$$

where

$$\gamma_v = a - 2b\cos\frac{v\pi}{n+1} \quad \text{and} \quad X_{vk} = \frac{(-1)^{n-k}}{\sqrt{\frac{n+1}{2}}} \sin\frac{kv\pi}{n+1}$$

Similarly, it is easy to show that the transformation R*AR (where R* is the transpose of R) where

$$R \equiv \begin{bmatrix} X_{11}I_n & \cdots & X_{v1}I_n & \cdots & X_{n1}I_n \\ X_{12}I_n & \cdots & X_{v2}I_n & \cdots & X_{n2}I_n \\ & & & & \\ X_{1n}I_n & \cdots & X_{vn}I_n & \cdots & X_{nn}I_n \end{bmatrix}$$ with X_{vk} as given

above reduces A to the quasi-diagonal form

$$\begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \cdots & \\ & & & M_n \end{bmatrix}$$

where $$M_v = D - 2b\cos\frac{v\pi}{n+1} I_n$$

Since M_v is equal to D with a change of the diagonal element, we have that the (k,l) element of the (i,j) block of e^A is

$$e^A_{i,j;k,l} = \sum_{v=1}^{n} X_{vi}X_{vj} \sum_{p=1}^{n} \exp\left(a - 2b\cos\frac{v\pi}{n+1} - 2b\cos\frac{p\pi}{n+1}\right) X_{pk}X_{pl}$$

and

$$A^{-1}_{i,j;k,l} = \sum_{n=1}^{n} X_{vi}X_{vj} \sum_{p=1}^{n} \left(a - 2b\cos\frac{v\pi}{n+1} - 2b\cos\frac{p\pi}{n+1}\right)^{-1} X_{pk}X_{pl}$$

where

$$X_{vk} = \frac{(-1)^{n-k}}{\sqrt{\frac{n+1}{2}}} \sin \frac{kv\pi}{n+1}$$

In our system $a = -\sigma$ and $b = \mu^{(0)}/5$ so the time constants, i.e., the arguments in each of the exponentials appearing in the solution for e^{At} are of the form

$$-\sigma t - \frac{2\mu^{(0)}}{5} t \left[\cos \frac{v_i \pi}{n+1} + \cos \frac{v_j \pi}{n+1}\right]$$

which takes on its smallest absolute value for $v_i = v_j = n$. Thus the motion of the system is bounded by

$$\exp - (\sigma - \frac{4}{5} \mu^{(0)} \cos \frac{\pi}{n+1}) t$$

The number n is the square root of the number of nodes in the square lattice. This result shows that as $n \to \infty$ the system attains its steady state at a rate

$$\exp - (\sigma - \frac{4}{5} \mu^{(0)}) t$$

which agrees with simulation results for $n = 32$.

APPENDIX F

OPTIMAL FIXED MESSAGE BLOCK SIZE FOR COMPUTER COMMUNICATIONS

by W. W. Chu

(Preprint of IFIP Congress 71, Ljubljana, Yugoslavia, August 23-28, 1971)

OPTIMAL FIXED MESSAGE BLOCK SIZE FOR COMPUTER COMMUNICATIONS*

WESLEY W. CHU
Computer Science Department
University of California, Los Angeles, USA

In many computer communication systems, random length messages are partitioned into fixed size blocks for ease in data handling and memory management. When error detection and retransmission are used in the error control procedures, there is at least one acknowledgment delay associated with each transmitted message block. Thus, from an acknowledgment point of view, it is desirable to select the larger block size so as to yield the fewer acknowledgments per message. On the other hand, the larger message block has a higher error probability and so may result in more retransmissions and more acknowledgment delays than the shorter message block size. Further, due to the random length of messages, the last partitioned block usually cannot be entirely filled with messages and is filled with dummy information. The large block size has a larger amount of such wasted channel bandwidth; hence, it is desirable to partition the message into smaller block size. Thus there is a trade-off among acknowledgment delay, message error probability, and the wasted channel bandwidth due to the last unfilled partitioned block. A mathematical model is developed in this paper to determine the optimal message block size that maximizes channel efficiency. Using the model, the relationships among acknowledgment time, channel transmission rate, channel error characteristics (random error or burst error), average message length, and optimal block size are obtained and presented in graphs. The model and the graphs should be useful as a guide in the selection of the optimal fixed message block size for computer-communication systems.

1. INTRODUCTION

To increase utilization of computer capability and to share computer resources, remotely located computers and/or terminals may be connected with communication links. Such integrated computer communication systems allow many users to economically share data bases and computer software systems. These shared computing facilities can greatly increase our computing capacity. In the design and planning of such systems, communication problems between computers and terminals greatly influence system performance (e.g., inquiry-response delay) and overall system costs. Hence, computer communications become an integral part of the overall system design consideration. For example, Asynchronous Time Division Multiplexing [1] has been proposed for data communication to increase channel utilization and reduce communication costs. In this paper we address the problem of determining the optimal fixed message block size to improve efficiency in data communication systems. Kucera [2], Balkovic and Muench [3], and Kirlin [4] have studied the optimal message block size for the error detection and retransmission system that maximizes transmission efficiency. In this paper, we consider an additional important parameter — the average message (file) length — in determining the optimal message block size, which significantly effects the selection of the optimal message block size.

For economic and reliability reasons [5, 6], the error detection and retransmission scheme is used by many data communication systems. Using this error control technique, encoding and decoding circuits usually are required to operate on the message information to process redundant data. The receiving end checks the received message (together with the redundant data) and then generates an acknowledgment signal for the sender to indicate whether the message is correctly received. If the message is correctly received (positive acknowledgment), then the sender is permitted to send a new message. If the message is incorrectly received (negative acknowledgment), then the sender retransmits the same message, until a positive acknowledgment of that message is

*This research was supported in part by the Advanced Research Projects Agency of the Department of Defense Contract No. DAHC 15-69-C-0285, U. S. Atomic Energy Commission Contract No. AT (11-1) Gen - 10, Project 14 and U.S. Office of Naval Research, Research Program Office, Contract No. N00014-69-A-0200-4027, NR 048-129.

received by the sender.

The message outputs from a computer are usually in strings of characters or bursts. The message length may be different from one to another and can best be described by a probability distribution. For ease in data handling and memory management, the random message length is usually partitioned into several fixed size blocks. Due to the random length of the message, the last partitioned block usually cannot be entirely filled by the message and is filled with dummy information. From the acknowledgment point of view, it is desirable to select the largest possible block size. Since each message block requires at least one acknowledgment signal, the fewer the number of blocks needed for a message, the less the channel capacity required for acknowledgments. On the other hand, since the larger message block has a higher probability of error and also has a higher channel wastage due to the last unfilled partitioned block, it is desirable to select the smallest possible block size. Thus there is a tradeoff in selecting the optimal block size. The basic problem is: suppose the average message length, the message length distribution, the channel error characteristics, block overhead, and the acknowledgment overhead are known. What is the optimal message block size that minimizes the time wasted in acknowledgments, retransmissions, and the last unfilled block?

A mathematical model is developed in this paper to treat this problem. The model considers two types of error channels: 1) Random error channel; that is, the errors are generated in a statistically independent fashion and the error rate can be approximated as a linear function of the block size, and 2) Burst error channel; that is, the errors are generated in a statistically dependent fashion such as the noise produced by radio static or switching transients, and the error rate is a nonlinear function of block size. In general, error characteristics can be obtained only from actual measurements [7]. For a given average channel error rate, the performance of the burst error channel is, in general, better than that of the random error channel [8]. Using our mathematical model, the relationships among message length (assuming the messages are geometrically distributed), transmission rate, acknowledgment overhead, block overhead, and optimal block size are obtained and portrayed in graphs. These graphs and the model should be useful as a guide or tool in selecting optimal message block size for planning computer-communication systems.

2. ANALYSIS

The message length, L, is a random variable and can be described by a probability distribution $P_L(\ell)$ with average length $\bar{\ell}$ characters per message. When the message is partitioned into a fixed size block of B characters per block, the expected number of blocks per message is equal to

$$N(B) = \sum_{n=1}^{\infty} n \cdot P_L\{(n-1)B < \ell \leq nB\} \quad \ell = 1, 2, \ldots \quad (1)$$

The structure of a fixed message block consists of an address, b_1, in the front of the message block and a checking code, b_2, after the message block, as shown in Figure 1.

B = FIXED SIZE MESSAGE BLOCK

b_1 = ADDRESS

b_2 = CHECK BITS

$b = b_1 + b_2$ = BLOCK OVERHEAD

Fig. 1. Data Structure of a Fixed Size Message Block.

The overhead of such a block is then equal to $b = b_1 + b_2$ characters.* Thus, for a message block size of B characters, the whole block length is equal to B + b characters. Let E(B+b) be the probability that a block of B + b characters (a message block of B characters with a block overhead b characters) transmitted over a channel will have at least one error. We know that E(B+b) is dependent on both the error characteristics of the channel and the whole block size B+b. Clearly, a larger value of B+b and/or a noiser transmission channel yields a higher value of E(B+b). When error detection and retransmission are used for error control, there is a certain amount of acknowledgment delay, A, associated with each message block. In a half duplex transmission mode, the acknowledgment delay should also include the modem turn-around times if modems are used in the channel. Thus, the expected acknowledgment overhead for a message block

*We assume the units of characters for consistency and allow the reader to insert the number of bits per character to fit his implementation.

transmitted over a channel is equal to the first acknowledgment time plus the expected retransmission time and reacknowledgment time. Assumming that the probability of error of each message block during transmission is independent of transmission or retransmission, then mathematically the expected acknowledgment overhead for a message block of size B (or whole block size B+b) on a channel with a transmission rate R character/sec is

$$\bar{A}(B+b) = A + \sum_{i=1}^{\infty} [E(B+b)]^i \cdot \left(A + \frac{B+b}{R}\right) \qquad (2)$$

The expected wasted time due to acknowledgement and retransmission in transmitting a message in fixed sized blocks, $W_1(B)$, is equal to the expected number of blocks (of size B+b) per message multiplied by the expected acknowledgment overhead of each block. Thus,

$$W_1(B) = N(B) \cdot \bar{A}(B+b)$$

$$= N(B) \cdot \left\{A + \sum_{i=1}^{\infty} [E(B+b)]^i \cdot \left(A + \frac{B+b}{R}\right)\right\} \qquad (3)$$

Since $0 \le E(B+b) \le 1$, $B > 0$, and $b > 0$,

$$\sum_{i=1}^{\infty} [E(B+b)]^i = \frac{E(B+b)}{1-E(B+b)} \qquad (4)$$

Substituting (4) into (3) yields

$$W_1(B) = N(B) \cdot \left\{A + \frac{E(B+b)}{1-E(B+b)} \cdot \left(A + \frac{B+b}{R}\right)\right\} \qquad (5)$$

The expected wasted time due to block overhead and the last unfilled partitioned block in transmitting a message in fixed sized blocks, $W_2(B)$, is equal to the difference between the time to transmit the blocked message and the unblocked message.

Thus

$$W_2(B) = N(B) \cdot \frac{B+b}{R} - \frac{\bar{\ell}+b'}{R} \qquad (6)$$

where b' is the overhead for the unblocked message.

The total expected wasted time to transmit a message in blocks, W(B), is equal to the sum of (5) and (6). Thus

$$W(B) = N(B) \cdot \left\{A + \frac{E(B+b)}{1-E(B+b)} \cdot \left(A + \frac{B+b}{R}\right) + \frac{B+b}{R}\right\} - \frac{\bar{\ell}+b'}{R} \qquad (7)$$

We wish to find the optimal block size B^o that minimizes (7); that is,

$$W(B^o) = \min_B \left\{N(B) \cdot \left[A + \frac{E(B+b)}{1-E(B+b)} \cdot \left(A + \frac{B+b}{R}\right) + \frac{B+b}{R}\right] - \frac{\bar{\ell}+b'}{R}\right\} \qquad (8)$$

Let us assume that the message length is geometrically distributed;* that is, $P_L(\ell) = pq^{\ell-1}$, $\ell = 1, 2, 3, \ldots$, with average message length $\bar{\ell} = p^{-1}$ and where $p+q = 1$. The average number of blocks per message in this case can be computed from (1) and is equal to

$$N(B) = \sum_{n=1}^{\infty} n \cdot P_L\{(n-1)B < \ell \le nB\} \qquad \ell = 1, 2, \ldots$$

$$= \sum_{n=1}^{\infty} n \cdot \sum_{\ell=(n-1)B+1}^{nB} pq^{\ell-1}$$

$$= \sum_{n=1}^{\infty} n \cdot (1-q^B)(q^B)^{n-1}$$

$$= (1-q^B)^{-1} \qquad (9)$$

Substituting (9) into (7), we have

$$W(B) = (1-q^B)^{-1}\left[A + \frac{E(B+b)}{1+E(B+b)} \cdot \left(A + \frac{B+b}{R}\right) + \frac{B+b}{R}\right] - \frac{1}{pR} - \frac{b'}{R} \qquad (10)$$

To minimize W(B), we take the derivative of (10) and set it equal to zero,

$$\frac{\partial W(B)}{\partial B} = 0,$$

or

$$(q^{-B}-1)\left[\frac{1}{AR} + \left(1 + \frac{B+b}{AR}\right)\frac{E'(B+b)}{1-E(B+b)}\right] + \left[1 + \frac{B+b}{AR}\right] \ln q = 0 \qquad (11)$$

where $E'(B+b)$ is the first derivation of $E(B+b)$.

It can be shown that the second derivative of W(B) (Eq. (10)) with respect to B that satisfies (11) is positive. Hence, solution of (11) yields the minimum W(B). Due to the complexity of (11), a closed form solution of B^o cannot be obtained. Thus, numerical techniques are used to solve (11) for B^o.

Let us study the behavior of Equation (11) and denote

$$X(B) = (q^{-B}-1)\left[\frac{1}{AR} + \left(1 + \frac{B+b}{AR}\right)\frac{E'(B+b)}{1-E(B+b)}\right]$$

*Measurements collected from several time sharing systems revealed that the message length output from these computers can be approximated by a geometrical distribution [9].

and

$$Y(B) = -\left[1 + \frac{B+b}{AR}\right] \cdot \ell n\, q$$

thus

$$X(B) - Y(B) = 0 \qquad (12)$$

First, let us evaluate X(B) and Y(B) at B = 0,

$$X(B)\big|_{B=0} = 0$$

and $$Y(B)\big|_{B=0} = \left[1 + \frac{b}{AR}\right] |\ell n\, q| \qquad 0 \le q \le 1$$

Thus, Y(0) ≥ X(0). Further, if the slope of X(B) is greater than the slope of Y(B) for all B ≥ 0, then X(B) intersects Y(B) at some B, 0 < B < ∞, as shown in Figure 2. This implies

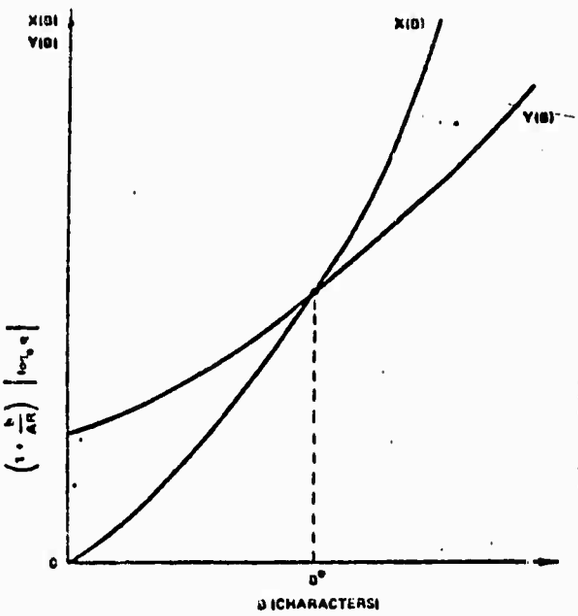


Fig. 2. $\frac{dX(B)}{dB} \ge \frac{dY(B)}{dB}$ and Y(0) > X(0)

that W(B) is a convex function of B. Hence, the optimal block size B located by the numerical technique attains a global optimal.

Next, differentiating X(B) and Y(B) with respect to B, we have

$$\frac{dX(B)}{dB} = (q^{-B} - 1)\left\{\frac{1}{AR} \cdot \frac{E'(B+b)}{1-E(B+b)} + \left(1 + \frac{B+b}{AR}\right)\frac{[1-E(B+b)]E''(B+b) + [E'(B+b)]^2}{[1-E(B+b)]^2}\right\} + \frac{1}{AR}\left[1 + (AR+B+b)\frac{E'(B+b)}{1-E(B+b)}\right] q^{-B} \cdot |\ell n\, q| \qquad (13)$$

$$\frac{dY(B)}{dB} = \frac{B+b}{AR} \cdot |\ell n\, q| \qquad (14)$$

where E''(B+b) is the second derivative of E(B+b).

Since the error probability increases as B increases, E'(B+b) ≥ 0. Also since 0 < E(B+b) ≤ 1, the second term of (13) is greater than $\frac{dY(B)}{dB}$. Further, if E''(B+b) ≥ 0, then the first term of (13) is positive. Thus, $\frac{dX(B)}{dB} \ge \frac{dY(B)}{dB}$ for all B. Therefore, E''(B+b) > 0 is a *sufficient* condition to assure the convexity of W(B). The physical meaning of E''(B+b) ≥ 0 implies that E(B+b) is a convex function of B+b. Comparing Equations (13) and (14), we know that even if E''(B+b) < 0 for some B, $\frac{dX(B)}{dB} \ge \frac{dY(B)}{dB}$ might still be satisfied. Therefore, E''(B+b) ≥ 0 is not a necessary condition for convexity of W(B).

In the following we shall analyze the optimal message block size for two types of error channels: random error channel and burst error channel.

Random Error Channel

For a random error channel, the probability that a block of B+b characters transmitted over a channel will have at least one error is

$$E(B+b) = 1 - (1-K)^{B+b}$$

$$= 1 - \left[1 - (B+b)K + \frac{(B+b)(B+b-1)K^2}{2!} - \frac{(B+b)(B+b-1)(B+b-2)}{3!}K^3 + \cdots\right] \qquad (15)$$

where K equals average channel character error rate. For most practical systems, (B+b)·K ≪ 1 (e.g., (B+b)K = 1000·10⁻⁴ = 0.1). Hence, (15) can be approximated as

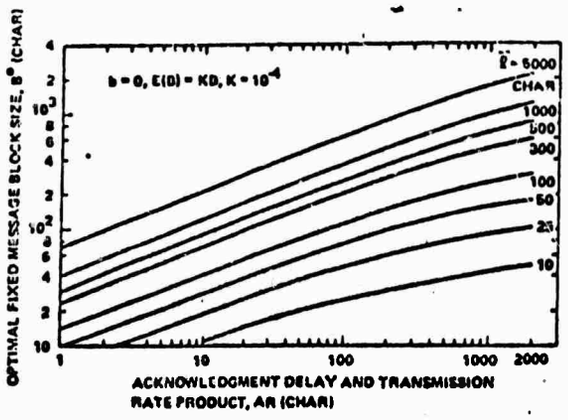
$$E(B+b) \doteq (B+b)K \qquad (16)$$

which is linearly proportional to the total block size B+b. Physically (16) implies that E(B+b) is approximately equal to the expected number of error characters in the block during transmission. The first and second derivatives of E(B+b) equal to E'(B+b) = K and E''(B+b) = 0.

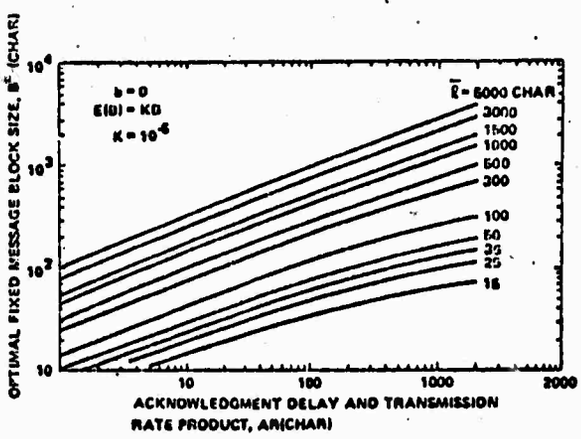
To determine the optimal message block size, we substitute E(B+b) ≐ K(B+b) and E'(B+b) = K into (11). We have

$$(q^{-B} - 1)\left[\frac{1}{AR} + \left(1 + \frac{B+b}{AR}\right)\frac{K}{1-K(B+b)}\right] + \left[1 + \frac{B+b}{AR}\right] \cdot \ell n\, q = 0 \qquad (17)$$

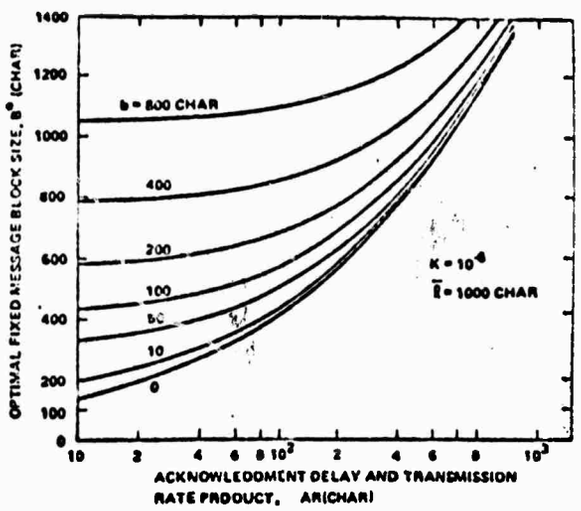
Since $E''(B+b) = 0$, $W(B)$ is a convex function of B. Based on the values of the product of acknowledgment delay and transmission rate AR, K, and $\bar{\ell} = (1-q)^{-1}$, the Newton-Raphson's Iterative Method [10] can be used to solve (17) for the optimal average block size B^o. The iteration is terminated when the improvement on W(B) from each new B is less than 10^{-4} seconds and the difference between the value of new B and its previous B is less than 0.1 characters. The optimal message block size for selected ranges of AR, K, $\bar{\ell}$ and b are portrayed in Figure 3.



(3a) $K = 10^{-4}$, $b = 0$



(3b) $K = 10^{-6}$, $b = 0$



(3c) $K = 10^{-6}$, $b \neq 0$

Fig. 3. Optimal Fixed Message Block Size for Random Error Channels when Message Length is Geometrically Distributed.

Burst Error Channel

In a burst error channel, the error tends to cluster rather than be evenly distributed over the messages. For example, the noise produced by radio static or switching transients may cause such burst errors. Two measured burst error channel characteristics [7] are shown in Figure 4.

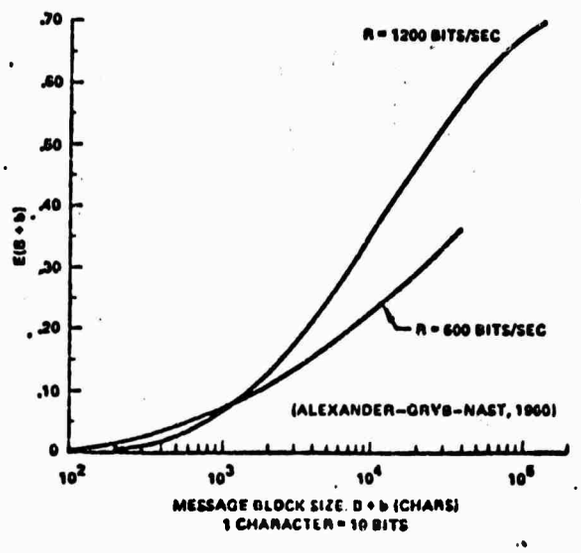


Fig. 4. Burst Error Channel Characteristics (Alexander-Gryb-Nast)

In order to express these curves mathematically, a curve fitting technique [10] is used to represent $E(B+b)$ as a polynomial. A good fit with mean of the square errors = 0.156×10^{-4} for the 600 bits/sec channel is

$$E_1(B+b) = 0.140 - 0.160\,[\log_{10}(B+b)] + 0.045\,[\log_{10}(B+b)]^2 \qquad (18)$$

In the same manner, a good fit with means of the square errors = 0.76×10^{-5} for the 1,200 bits/sec channel is

$$E_2(B+b) = -0.145 + 0.155[\log_{10}(B+b)] - 0.108[\log_{10}(B+b)]^2 + 0.121\times10^{-1}[\log_{10}(B+b)]^3 + 0.243\times10^{-2}[\log_{10}(B+b)]^4 - 0.146\times10^{-1}[\log_{10}(B+b)]^5 + 0.266\times10^{-3}[\log_{10}(B+b)]^6 - 0.273\times10^{-4}[\log_{10}(B+b)]^7 - 0.162\times10^{-4}[\log_{10}(B+b)]^8 + 0.230\times10^{-5}[\log_{10}(B+b)]^9 .$$

$$\text{for } 200 \le B+b \le 2\times10^5 \text{ chars} \qquad (19)$$

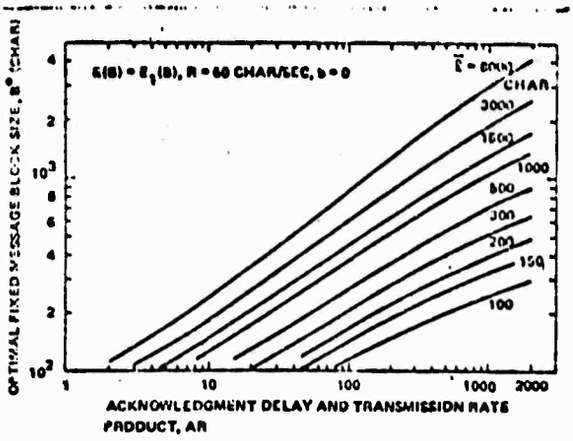
The $(B+b)$'s in (18) and (19) are in characters. The original measured results are in bits. Here, for consistency, we represented them in characters with a scale of one character equaling ten bits.

From (18), we know that $E_1''(B+b) > 0$, for all $B+b$. Thus, the $W(B)$ of $E_1(B+b)$ is a convex function of B. From (19), $E_2''(B+b) < 0$ for some B. Hence, we need to compute and compare its $\frac{dX(B)}{dB}$ and $\frac{dY(B)}{dB}$. We find that for the range of B of interest, i.e., $0 \le B \le 10^5$ characters, $\frac{dX(B)}{dB} \ge \frac{dY(B)}{dB}$. Therefore, $W(B)$ for $E_2(B+b)$ is also a convex function of B.

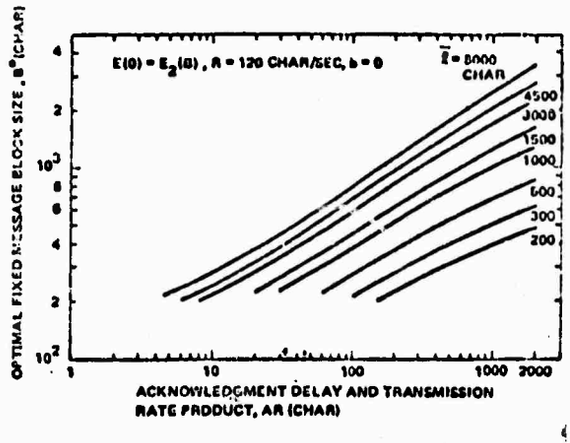
In the same manner as in the random error case, we substitute (18) into (17) and then (19) into (17) and use numerical techniques to solve for B^o for each of the transmission rates. The relationship among the optimal block size,* average message length, b, and AR's for $E_1(B+b)$ and $E_2(B+b)$ are portrayed in Fig. 5.

When $\frac{dX(B)}{dB} \ge \frac{dY(B)}{dB}$ are not satisfied for some regions of B, numerical results might lead to a local optimal. In this case, numerical search should be performed in these regions to locate the local optimals of each region. The global optimal block size B^o can then be selected from these local optimals.

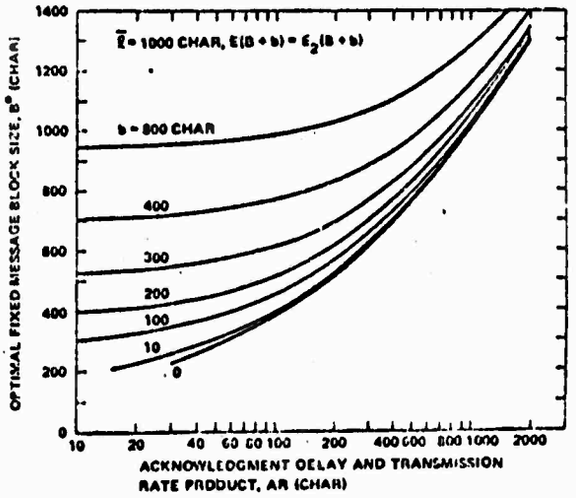
*Note $E_1(B+b)$ and $E_2(B+b)$ are represented in characters with a scale of one character equals ten bits. Should the message not agree with the scale as that of $E_1(B+b)$ and $E_2(B+b)$, the message must be converted into the same scale (one character equals ten bits) before using Fig. 5 to determine the optimal block size.



(5a) $E(B+b)=E_1(B+b)$, R=600 bits/sec, b=0



(5b) $E(B+b)=E_2(B+b)$, R=1200 bits/sec, b=0



(5c) $E(B+b)=E_2(B+b)$, R=1,200 bits/sec, $b\neq0$

Fig. 5. Optimal Fixed Message Block Size for Burst Error Channels when Message Length is Geometrically Distributed.

3. DISCUSSION OF RESULTS

The optimal block size B^o for a random error channel with selected average message length, block overhead, and AR's is presented in Fig. 3. For a specific random error channel, the optimal block size increases with the AR. This agrees with our intuition, i.e., if A and/or R is increasing, (note, A is independent of R), then we should partition the message into larger blocks so as to reduce the number of blocks per message.

For a given AR and average message length, the optimal block size is larger for a smaller error rate channel than for a larger error rate channel.

For a burst error channel, we approximated its error characteristics by a polynomial. In the same manner as in the random error channel case, we substituted $E_1(B+b)$ and $E_2(B+b)$, respectively, into (17) and numerically solved for the optimal message block size. The B^o's for selected average message lengths and selected AR's of $E_1(B+b)$ and $E_2(B+b)$ are presented in Fig. 5. For small AR values, the behavior of B^o is similar to that of a low error rate random error channel. For large AR values, the behavior of B^o is similar to that of a high error rate random error channel. This phenomena is due mainly to the fact that the burst error characteristics have a nonlinear effect with block size.

The W(B) for a random error channel is always a convex function. The W(B)'s for a burst error channel with error characteristics $E_1(B+b)$ and $E_2(B+b)$ are also convex functions. In general, however, the W(B) for a burst error channel is not necessarily a convex function. A few typical W(B)'s are shown in Fig. 6.

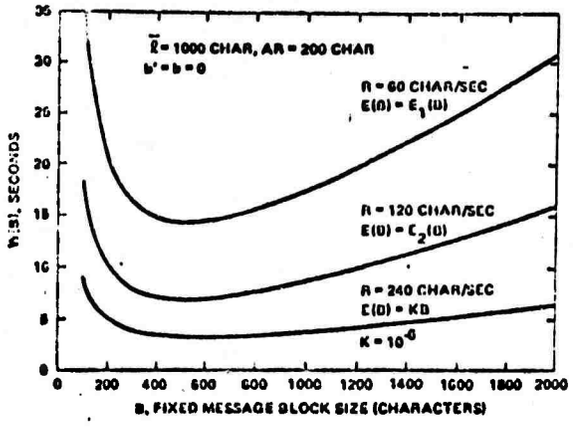


Fig. 6. Several Typical W(B) vs B

We noted that the W(B)'s are rather insensitive around B^o. For a given $\bar{\ell}$, AR, and channel error characteristic, the W(B) increases as the block overhead b increases as shown in Fig. 7.

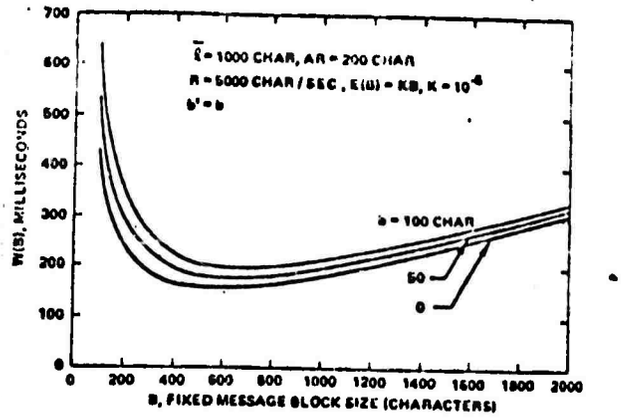


Fig. 7. Effect of W(B) with Block Overhead, b.

From Figures 3c and 5c, we noted that $B^o(b\neq 0)$ increases as b increases, and the amount of difference between $B^o(b=0)$ and $B^o(b\neq 0)$ decreases as AR increases. From Fig. 6, we noted that the W(B)'s are rather insensitive around B^o. Therefore, for a system that has a small block overhead b, and a large AR, then the optimal block size with block overhead is approximately equal to that without block overhead, that is, $B^o(b\neq 0) \doteq B^o(b = 0)$.

4. EXAMPLE

Consider the planning of a computer network that consists of many computers and/or terminals. These computers are remotely located and communicate from each other via communication channels. A wideband channel that can transmit 5,000 characters per second is used between each pair of computers. The channel has burst error channel characteristics similar to $E_2(B+b)$ as shown in Fig. 4. For reliability in information transfer, error detection and retransmission are employed in the system. Further, the message is partitioned into fixed size blocks for ease in data handling and memory management. The block overhead, b, is approximately equal to ten characters. The acknowledgment time for each block is about 40 milliseconds. The message length of the computer output can be approximated as geometrically distributed. We would like to consider the optimal block sizes for: 1) average message length equal to 500 characters, and 2) average message length equal to 1,000 characters.

The acknowledgment-transmission rate product is equal to $AR=40\times10^{-3}\times5\times10^{3}=200$ characters. Solving Equation (11) numerically, we find the optimal block size for $\bar{\ell} = 500$ characters is 376 characters, and from Fig. 5c, the optimal block size for $\bar{\ell} = 1,000$ characters is 527 characters. Comparing with their optimal block sizes for b = 0, the differences between the $D^{o}(b\neq0)$ and $D^{o}(b=0)$ are within ten characters.

5. ACKNOWLEDGMENT

The author wishes to thank Leo Liang of UCLA for his programming assistance in the numerical solution for the optimal message block size.

REFERENCES

[1] W. W. Chu, A Study of Asynchronous Time Division Multiplexing Technique for Time Sharing Computer Systems, Proceedings of FJCC, Vol. 35, (1969) pp. 669-678.

[2] J. J. Kucera, Transfer Rate of Information Bits, Computer Design (June 1968) pp. 56-59.

[3] M. D. Balkovic and P. E. Muench, Effect of Propagation Delay, Caused by Satellite Circuits, on Data Communication Systems that Use Block Retransmission for Error Correction, Conference Record IEEE Conference on Communications, Boulder, Colorado, (June 9-11, 1969) pp. 29-31 to pp. 29-36.

[4] R. L. Kirlin, Variable Block Length and Transmission Efficiency, IEEE Transactions on Communication Technology, Vol. Com. -17, No. 3 (June 1969) pp. 350-354.

[5] R. L. Townsent and R. N. Watts, Effectiveness of Error Control in Data Communication Over the Switched Telephone Network, BSTJ, (November 1964) pp. 2611-2638.

[6] S. Y. Tong, A Survey of Error Control Techniques on Telephone Channels, Proceedings of the 1970 National Electronics Conference, Chicago (Dec. 7-9, 1970) pp. 462-467.

[7] A. A. Alexander, R. M. Gryb, and D. W. Nast, Capabilities of the Telephone Network for Data Transmission, BSTJ (may 1960) pp. 431-476.

[8] E. N. Gilbert, Capacity of a Burst-Noise Channel, BSTJ (March 15, 1960) pp. 1253-1265.

[9] E. Fuchs and P. E. Jackson, Estimates of Distributions of Random Variables for Certain Computer Communications Traffic Model, CACM, Vol. 13, No. 12 (Dec. 1970) pp. 752-757.

[10] R. W. Hamming, Numerical Methods for Scientists and Engineers, McGraw-Hill Book Company, (1962).

APPENDIX G

ON NON-BLOCKING SWITCHING NETWORKS

by D. G. Cantor

ON NON-BLOCKING SWITCHING NETWORKS *

DAVID G. CANTOR

Abstract

A switching network may be informally described as a collection of single-pole, single-throw switches arranged so as to connect a set of terminals called inputs to another set of terminals called outputs. It is non-blocking if, given any set of connections from some of the inputs to some of the outputs, and an idle input terminal x and idle output terminal y, then it is possible to connect x to y without disturbing any of the existing connections. Denote by $\sigma(a,b)$ the minimal number of switches necessary to connect a inputs to b outputs using a non-blocking network. We are interested in studying the growth of $\sigma(a,a)$ as $a \to \infty$. Results of C. Clos show that $\sigma(a,a) \leq C\ ae^{2\sqrt{\log a \cdot \log 2}}$. We show that $\sigma(a,a) \leq 8a(\log_2 a)^2$.

---

*This work was support in part by the Advanced Research Projects Agency, Department of Defense Contract DAHC-15-69-C-0285. The author would also like to thank the National Science Foundation GP #13164 and the Sloan Foundation for support while writing this paper.

ON NON-BLOCKING SWITCHING NETWORKS

DAVID G. CANTOR*

1. Introduction.

A Network N consists of a graph G; two sets of vertices of G, denoted A and B and called, respectively, the (sets of) inputs and outputs; and a set P of paths of G. Each path in P connects an input to an output and meets no other inputs or outputs. We write $N = (G,A,B,P)$. A state of N is a subset S of P such that no two paths in G have a common vertex. A state S defines a bijection f_S from a subset of A to a subset of B as follows: Suppose $p \in S$ and p connects $x \in A$ to $y \in B$; put $f_S(x) = y$, and repeat this for each path in S. We shall say that a path p of G is admissible if $p \in P$. If x is a vertex of G we shall say that x is busy (in the state S) if x lies on a path $p \in S$; otherwise we shall say that x is idle (in the state S). If x is an input of G and y is an output of G, we shall say that x has access to y (in the state S) if there exists a path $p \in P$ connecting x to y and such that $S \cup \{p\}$ is a state.

A network $N = (G,A,B,P)$ may be interpreted as a switching device; under this interpretation, the elements of A are considered as input terminals, the elements of B are considered as output terminals, and the edges of G are considered as single-pole, single-throw

switches which are normally open. Then a path p, which connects $x \in A$ to $y \in B$ may be thought of as a sequence of switches which, when closed, connect x to y. The state S yields a collection of switches (all edges on any path in S) which, when closed, connect inputs to outputs as described by the function f_S.

The network $N = (G,A,B,P)$ is said to be <u>non-blocking</u> if given any state S of N and idle vertices $x \in A$, $y \in B$, then x has access to y in the state S. In terms of the switching network interpretation mentioned above, this means that if x and y are idle input and output terminals, respectively, then it is possible to establish a connection between them without disturbing the existing connections.

From now on, all the networks we study will have disjoint inputs and outputs (i.e. $A \cap B = \emptyset$).

Given positive integers a and b we are interested in finding those non-blocking networks $N = (G,A,B,P)$ with $|A| = a$, $|B| = b$ for which the number of edges of G is minimal. We shall denote this number by $\sigma(a,b)$. In terms of switching networks, this amount to finding non-blocking networks using a minimal number of switches. An obvious non-blocking network with a inputs and b outputs is the network whose graph is the complete bipartite graph on vertex sets A and B with $|A| = a$ and $|B| = b$. In this graph the set of vertices

is $A \cup B$ and there is an edge connecting each vertex in A to each vertex in B. The set P consists of all paths consisting of exactly one edge. Thus P has ab elements. In the switching network interpretation, this amounts to an a by b crossbar switch. When the names of the sets A and B are unimportant, we shall denote this network by C_{ab}. The network C_{ab} shows that $\sigma(a,b) \leq ab$.

It was Clos [2] who showed that $\sigma(N,N) < N^2$ for all large N. His methods, which will be described later, show that $\sigma(N,N) \leq C\, N e^{2\sqrt{(\log N)\cdot(\log 2)}}$. We will show that $\sigma(N,N) \leq 8N(\log_2 N)^2$. We do not attempt to obtain the smallest possible constant multiplier, for it is not clear that the exponent 2 can not be reduced. In the opposite direction, an elementary argument shows that $\sigma(N,N) > C\, N \log_2 N$, and nothing stronger is known.

The author would like to acknowledge many stimulating discussions with Professors B. Gordon and C. B. Tompkins.

2. Constructions.

We shall say that networks $N = (G,A,B,P)$ and $N' = (G',A',B',P')$ are isomorphic (or equivalent) if there exists a graph isomorphism μ of G onto G' such that $\mu(A) = A'$, $\mu(B) = B'$, and $\mu(P) = P'$. It is clear that the property of being non-blocking is preserved under isomorphism.

If $N = (G,A,B,P)$ is a network, we define its transpose N' to be the network $N' = (G,B,A,P)$; clearly $N'' = N$.

If G is a graph and C is a set, we define the graph $G \times C$ to be the graph whose vertices are the ordered pairs (x,c) with x a vertex of G and $c \in C$; $((x_1,c_1),(x_2,c_2))$ is an edge of $G \times C$ if $c_1 = c_2$ and (x_1,x_2) is an edge of G. If p is a path in G whose vertices, in order, are $x_0,x_1,\ldots,x_n$ then by $p \times c$ we mean the path in $G \times C$ whose vertices are $(x_0,c),(x_1,c),\ldots,(x_n,c)$. The product $C \times G$ is defined similarly.

Now suppose $L_i = (G_i,A_i,B_i,P_i)$ $(i = 1$ or $2)$ are networks; we are going to define the network product $L_1 \times L_2$. We shall denote this product by $N = (H,C,D,Q)$. Put $C = A_1 \times A_2$ and $D = B_1 \times B_2$. The graph H is obtained from the two graphs $G_1 \times A_2$ and $B_1 \times G_2$ by identifying the vertices in $B_1 \times A_2$, which appear in both graphs. All admissable paths $q \in Q$ of N are obtained as follows: Let $p_i \in P_i$ be an admissible path connecting $x_i \in A_i$ to $y_i \in B_i$ $(i = 1$ or $2)$. Then $p_1 \times x_2$ ends in the vertex (y_1,x_2) which is the first vertex of $y_1 \times p_2$. The path $q = (p_1,p_2)$ is defined to be the path obtained from the paths $p_1 \times x_2$ and $y_1 \times p_2$ by concatenating them and identifying the common vertex (y_1,x_2). Note that this maps $P_1 \times P_2$ onto Q.

In the switching network interpretation this construction amounts

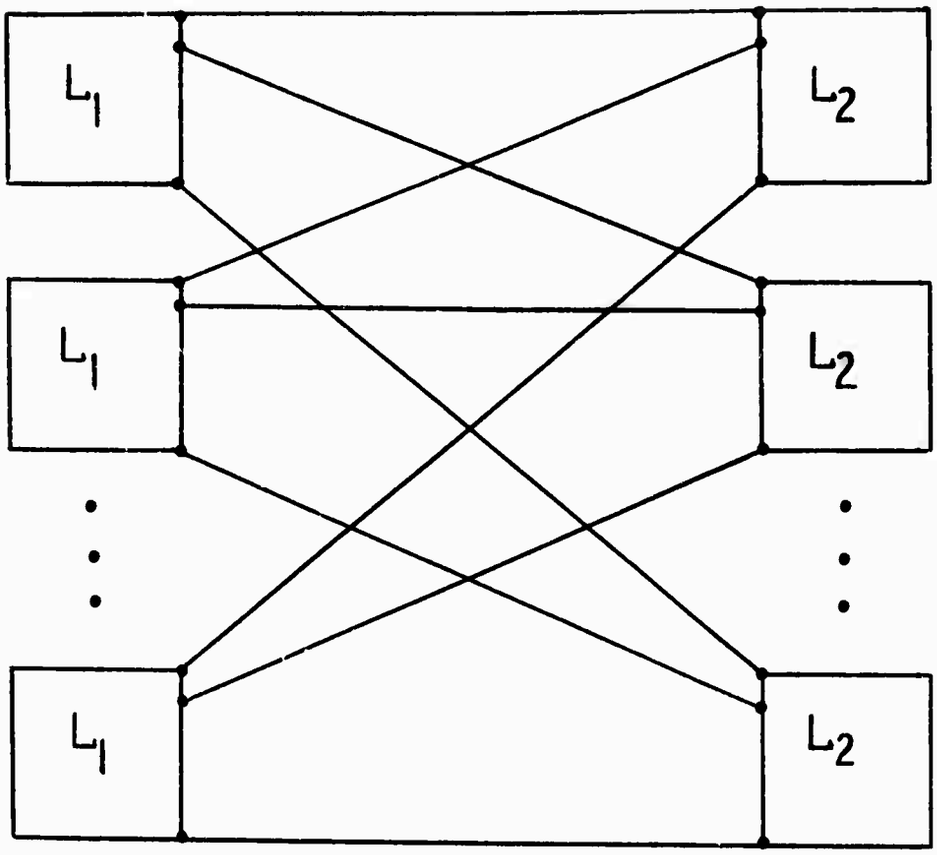
to taking $|A_2|$ copies of L_1 and $|B_1|$ copies of L_2, and connecting the outputs of each of the copies of L_1 to the inputs of all of the copies of L_2 (see Diagram 1).

Let a_i, b_i, c, d denote, respectively, the cardinalities of A_i, B_i, C, D, and let g_i, h denote, respectively the number of edges of G_i and H. The following relationship between two by two matrices is easily verified

$$(1) \qquad \begin{pmatrix} a_1 & 0 \\ g_1 & b_1 \end{pmatrix} \begin{pmatrix} a_2 & 0 \\ g_2 & b_2 \end{pmatrix} = \begin{pmatrix} c & 0 \\ h & d \end{pmatrix} .$$

If L_1 is isomorphic to M_1 and L_2 is isomorphic to M_2 it is easy to verify that $L_1 \times L_2$ is isomorphic to $M_1 \times M_2$. Furthermore $(L_1 \times L_2)' = L_2' \times L_1'$. Finally, we have associativity: $(L_1 \times L_2) \times L_3 = L_1 \times (L_2 \times L_3)$; we will usually write simply $L_1 \times L_2 \times L_3$. We will abbreviate the k-fold product $L \times L \times \cdots \times L$ by L^k.

We also define a triple product of the three networks $L_i = (G_i, A_i, B_i, P_i)$ $(i = 1,2,3)$ when $|B_1| = |A_3|$. Let τ be a bijection from A_3 onto B_1; the triple product of L_1, L_2, L_3 depends upon the choice of τ and will be denoted by $[L_1, L_2, L_3]_\tau$. (In many cases L_3 will be L_1' and in such cases we will choose τ to be the identity map. In any case those properties of the triple



The lines do <u>not</u> represent edges; they connect output vertices of L_1 to the input vertices of L_2 with which they are identified.

Figure 1. $L_1 \times L_2$

product which we will use will be independent of the choice of τ and we will frequently write $[L_1,L_2,L_3]$ instead of $[L_1,L_2,L_3]_\tau$.) Suppose then that $N = (H,C,D,Q)$ is $[L_1,L_2,L_3]_\tau$.

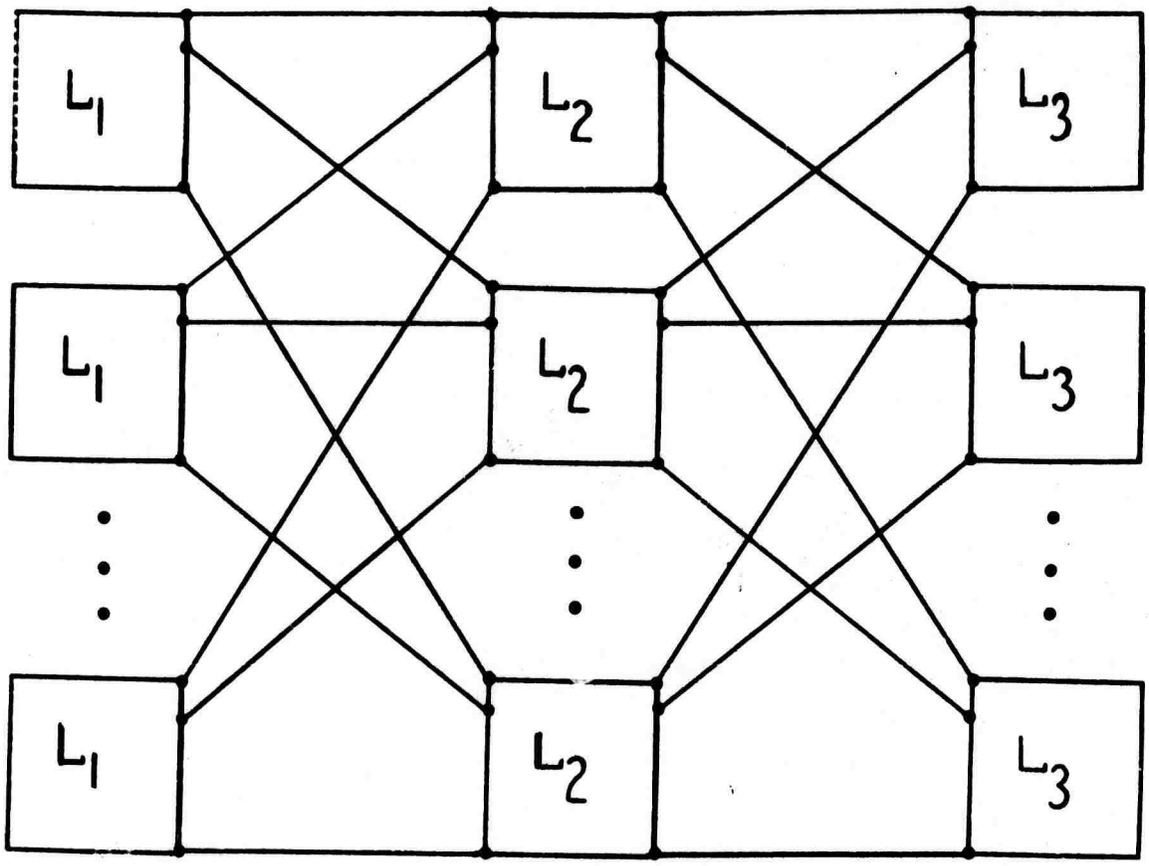
We put $C = A_1 \times A_2$ and $D = B_3 \times B_2$; H is defined as the graph obtained from the three graphs $G_1 \times A_2$, $B_1 \times G_2$, and $G_3 \times B_2$, by identifying $B_1 \times A_2$ in $G_1 \times A_2$ with $B_1 \times A_2$ in $B_1 \times G_2$, and by identifying $A_3 \times B_2$ in $G_3 \times B_2$ with $\tau(A_3) \times B_2 = B_1 \times B_2$ in $B_1 \times G_2$. The admissible paths $q \in Q$ are obtained in the following way: Let p_i be an admissible path of L_i connecting $x_i \in A_i$ to $y_i \in B_i$ $(i = 1,2,3)$ and suppose $\tau(x_3) = y_1$. Then $p_1 \times x_2$ ends at (y_1,x_2); $y_1 \times p_2$ begins at (y_1,x_2) and ends at (y_1,y_2); and $p_3 \times y_2$ begins at $(x_3,y_2) = (y_1,y_2)$. The path q is obtained by concatenating $p_1 \times x_2$, $y_1 \times p_2$, $p_3 \times y_2$ and identifying the vertices common to two segments of q.

Note that $[L_1,L_2,L_3]$ is, in general, different from $L_1 \times L_2 \times L_3$ (see Diagram 2).

The following is easily verified; we omit the proof.

THEOREM 2.1. Suppose $L_i = (G_i,A_i,B_i,P_i)$ $(i = 1,2,3,4,5)$ are networks. Suppose τ_1 is a bijection of A_5 onto B_1. Then

$$[L_1,[L_2,L_3,L_4]_{\tau_1},L_5]_{\tau_2} = [L_1 \times L_2,L_3,L_4 \times L_5]_\tau ,$$



The lines do <u>not</u> represent edges; instead they connect vertices which are to be identified.

Figure 2. $[L_1, L_2, L_3]$

where τ_3 is the bijection from $A_4 \times A_5$ to $B_1 \times B_2$ given by $\tau(a_4,a_5) = (\tau_2(a_5), \tau_1(a_4))$.

3. The Clos method and some variations.

The basic method, due to Clos [2] and quoted by Beneš [1] may be stated as the

THEOREM (Clos). *Suppose* $L = (G,A,B,P)$ *is non-blocking and* $s \geq 2r - 1$. *Then* $N = [C_{rs},L,C_{sr}]$ *is non-blocking.*

This is a special case of the following more general

THEOREM 3.1. *Suppose* $L_i = (G_i,A_i,B_i,P_i)$ $(i = 1,2,3)$ *are non-blocking, that* $|B_1| \geq |A_1| + |B_3| - 1$, *and that* $|B_1| = |A_3|$. *Then* $N_\tau = [L_1,L_2,L_3]_\tau$ *is non-blocking for any bijection* τ *of* A_3 *onto* B_1.

Proof. Suppose $N_\tau = (H,C,D,Q)$ is in state S, and that $x \in C, y \in D$ are idle. We must show there exists a path $q \in Q$ connecting x to y and having no common vertices with any path in S. Suppose $x = (u_1,u_2) \in A_1 \times A_2$ and $y = (v_3,v_2) \in B_3 \times B_2$. There are $|A_1|$ vertices of the form $(u,u_2) \in A_1 \times A_2$ and at most $|A_1| - 1$ of them are busy. Hence at most $|A_1| - 1$ of the $|B_1|$ vertices of the form $(y,u_2) \in B_1 \times A_2$ are busy and hence at least $|B_1| - |A_1| + 1$ of them are idle. Denote these vertices by

$(y_{i_1},u_2),(y_{i_2},u_2),\ldots,(y_{i_r},u_2)$, so that $r \geq |B_1| - |A_1| + 1$. Similarly, there are vertices $(z_{i_1},v_1),(z_{i_2},v_2),\ldots,(z_{i_s},v_s)$ in $A_3 \times B_2$ which are idle, and $s \geq |A_3| - |B_3| + 1$. The $r + s$ vertices $y_{i_1},y_{i_2},\ldots,y_{i_r}$, $\tau(z_{i_1}),\tau(z_{i_2}),\ldots,\tau(z_{i_s})$ all lie in B_1 and

$$\begin{aligned} r + s &\geq |B_1| - |A_1| + 1 + |B_1| - |B_3| + 1 \\ &\geq |B_1| + 1 + (|B_1| - |A_1| - |B_3| + 1) \\ &\geq |B_1| + 1 \ . \end{aligned}$$

So two of them must be the same. Now the y_{i_j} are all distinct and so are the $\tau(z_{i_j})$. Thus there must be a y_{i_j} equal to a $\tau(z_{i_k})$, say $y_{i_1} = \tau(z_{j_1})$. Since L_1 is non-blocking there is a path p_1 connecting u_1 to y_{i_1} and such that $p_1 \times u_2$ has no common vertices with any vertex in S. Similarly there is a path p_2 from u_2 to v_2 in P_2 such that $y_{i_1} \times p_2$ has no common vertices with any path in S, and there is a path p_3 from z_{j_1} to v_3 in P_3 such that $p_3 \times v_2$ has no vertex in common with any path in S. Let q be the concatenation of $p_1 \times u_2$, $y_{i_1} \times p_2$, and $p_3 \times v_2$ with the appropriate vertices identified. Then q connects x to y and $S \cup \{q\}$ is a state of N_τ.

REMARK 3.2. Suppose $a_i = |A_i|$, $b_i = |B_i|$ and g_i is the number of edges of G_i $(i = 1,2,3)$. It is easy to verify using (1) that $N = [L_1,L_2,L_3]$ has a_1a_2 inputs, b_2b_3 outputs and that its graph has $a_2g_1 + b_1g_2 + b_2g_3$ edges.

Clos [2] suggests using metworks which may be described as

$$[L,[L,[L,\ldots,[L,M,L'],L'],L'],\ldots,L']$$

where $L = C_{n,2n-1}$ and $M = C_{n,n}$. By Theorem 2.1, this is the same as $[L^t,M,(L')^t]$, where $L^t = L \times L \times L \times \cdots \times L$ (t times). He shows that this non-blocking network, which has n^{t+1} inputs and outputs, has

$$\frac{n^2(2n - 1)}{n - 1}\ [(5n - 3)(2n - 1)^{t-1} - 2n^t]$$

edges. This follows immediately from the above remark. It is easy to verify that a non-blocking network with N inputs and outputs, constructed by this method, will require at least $C_0\, e^{2\sqrt{\log N \cdot \log 2}}$ edges, where $C_0 > 0$ is a constant.

Suppose that L_{ab} denotes a network with a inputs, b outputs, and whose graph contains a minimal number of edges, namely $\sigma(a,b)$. Using two copies of L_{aa} shows that $\sigma(a,2a) \leq 2\sigma(a,a)$. By Theorem 3.1, $[L_{a,2a}, L_{a,a}, L_{2a,a}]$ is non-blocking and by Remark 3.2, it has $\leq a\sigma(a,2a) + 2a\sigma(a,a) + a\sigma(2a,a) \leq 6a\sigma(a,a)$ edges. Thus

$$\sigma(a^2,a^2) < 6a\sigma(a,a) \quad . \tag{2}$$

Iteration of (2) shows that $\sigma(N,N) \leq C\, N(\log N)^{\log_2 6}$. This result can be improved by considering $[L_{a,2a}, L_{a,2b}, L_{2a,a}]$; this network has ab inputs, 2ab outputs and its graph has $3b\sigma(a,2a) + 2a\sigma(b,2b)$ edges. This shows that

$$\sigma(ab,2ab) \leq 3b\sigma(a,2a) + 2a\sigma(b,2b) \quad . \tag{3}$$

Putting $a = b$ and iterating (3) shows that $\sigma(a,2a) \leq C\, a(\log_2 a)^{\log_2 5}$ and since $\sigma(a,a) \leq \sigma(a,2a)$ we find that

$$\sigma(N,N) \leq C\, N(\log_2 N)^{\log_2 5} \quad . \tag{4}$$

The exponent $\log_2 5$ can be decreased by choosing a and b differently. let $\alpha > 1$ and $\beta > 2$ be the real solutions of the simultaneous equations

(4) $$\left\{ \begin{array}{rcl} \alpha^{\beta-1} & = & 3 \\ (\alpha - 1)^{\beta-1} & = & 3/2 \end{array} \right\}$$

Numerical computation shows that $\alpha \simeq 2.37638$ and $\beta \simeq 2.26922$. Multiplying the second equation of (4) by $\alpha - 1$ and substituting from the first yields $2(\alpha - 1)^{\beta} = \alpha^{\beta} - 3$ or equivalently

(5) $$3(1/\alpha)^{\beta} + 2(1 - 1/\alpha)^{\beta} = 1 \ .$$

We now show that if $\mu(x) = (\log x)^{\beta}$, then $\mu(x)$ satisfies the functional equation

(6) $$\mu(z) = 3\mu(x) + 2\mu(y)$$

where $x = z^{1/\alpha}$ and $y = z/x$. Indeed,

$$\begin{aligned} 3\mu(x) + 2\mu(y) &= 3((\log z)/\alpha)^{\beta} + 2(\log z)^{\beta} (1 - 1/\alpha)^{\beta} \\ &= (\log z)^{\beta} \\ &= \mu(z) \end{aligned}$$

using (5).

Now $\sigma(x,2x)/x$ satisfies a functional inequality similar to (6) where x and y must be integers. It follows that for each $\varepsilon > 0$, there exists $C_{\varepsilon} > 0$ such that

$$\sigma(N,2N) \leq C_\varepsilon N(\log N)^{\beta+\varepsilon} .$$

For comparison, $\log_2 5 = 2.32193$.

4. The exponent is ≤ 2.

Suppose $L = (G,A,B,P)$ is a network (not necessarily non-blocking). We shall say that L is of type $T(m,n)$ if, given any state S of L and m idle inputs $x_1,x_2,\ldots,x_m$ of L, then each has access, in the state S, to at least n outputs of L.

LEMMA 4.1. *Suppose $L = (G,A,B,P)$ is of type $T(m,m+n-1)$ for $1 \leq m \leq k$, that M is a non-blocking network with c inputs and d outputs, and that $nd \geq a(c-1)$. Then $L \times M$ is of type $T(m,m+n'-1)$ for $1 \leq m \leq k$ where $n' = nd - a(c-1)$ and a is the number of inputs of L.*

Proof. Take $k \leq m$ idle inputs $z_1,z_2,\ldots,z_k$. Suppose, for example, that $z_1,z_2,\ldots,z_{k'}$ are of the form

$$(x_1,y_1),(x_2,y_1),\ldots,(x_{k'},y_1) ,$$

and $z_{k'+1},z_{k'+2},\ldots,z_k$ are of the form (x_h,y_i) where $i \geq 2$; here the x_j are inputs of L and the y_j are inputs of M. By hypothesis,

(x_1,y_1) has access to at least $n + k' - 1$ vertices of the form (u_j,y_j) where the u_j are outputs of L. Since M is non-blocking, these have access to all idle vertices of the form (u_j,v_k) where v_k is an output of M. There are $(n + k' - 1)d$ such vertices. However, as many as $(c - 1)a - (k - k')$ of these could be busy; this would be the case if all inputs of the form (x_h,y_i), where $i \geq 2$, other than $z_{k'+1}, z_{k'+2}, \ldots, z_n$ were busy. Thus z_1 has access to at least

$$(n + k' - 1)d - (c - 1)a + (k - k') \geq nd - (c - 1)a + k - 1$$

$$= n' + k - 1$$

output terminals of $L \times M$.

The following theorem provides the motivation for defining the notion $T(m,n)$.

THEOREM 4.2. Suppose M is a non-blocking network and L is a network with a inputs, b outputs, and of type $T(1,n)$. If $2n > b$, then $[L,M,L']$ is non-blocking.

The proof is similar to that of Theorem 3.1 and will be omitted.

Now choose an integer $k \geq 1$ and put $L_j = C_{2,2k} \times C_{2,2}^{j-1}$;

if $j \leq k$, then L_j has 2^j inputs, $k \cdot 2^j$ outputs, and inductively by Lemma 4.1, L_j is of type $T(1, 2^{j-1}(2k - j))$ and $T(2, 2^{j-1}(2k - j) + 1)$. Thus L_k is of type $T(2, k2^{k-1} + 1)$. Let M_k be obtained from L_k by omitting one input. Then M_k has $2^k - 1$ inputs $k \cdot 2^k$ outputs, is of type $T(1, k \cdot 2^{k-1} + 1)$, and its graph has no more edges than the graph of L_k. The associated matrix of L_k is

$$\begin{pmatrix} 2 & 0 \\ 4k & 2k \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 4 & 2 \end{pmatrix}^{k-1} = 2^k \begin{pmatrix} 1 & 0 \\ 2k^2 & k \end{pmatrix} .$$

Thus M_k has $2^k \cdot 2k^2$ edges and if N is any non-blocking network, then by Theorem 4.2, so is $[M_k, N, M_k']$. Thus putting, for example, $N = C_{2,2}$, we obtain a non-blocking network with $(2^{k+1} - 2)$ inputs and outputs whose graph has $\leq 2^{k+1}(4k^2 + 2k)$ edges. It is immediate that $\sigma(N,N) \leq 8N(\log_2 N)^2$ for all $N \geq 2$. It is not hard to see that the constant 8 could be considerably decreased, but the major open question is the value of the exponent.

REFERENCES

1. V. E. Benes, Mathematical Theory of Connecting Networks and Telephone Traffic, New York, Academic Press, 1965.

2. C. Clos, A Study of Non-Blocking Switching Networks, Bell System Tech. J. 32 (1953), pp. 406-424.

UNIVERSITY OF CALIFORNIA, LOS ANGELES
LOS ANGELES, CALIFORNIA 90024

APPENDIX H

DELAY IN COMMUNICATION AND COMPUTER NETWORKS

by L. Kleinrock

DELAY IN COMMUNICATION AND COMPUTER NETWORKS*

Leonard Kleinrock
Computer Science Department
University of California, Los Angeles, California 90024

I. INTRODUCTION

Delay in communication and computer networks has recently become a subject of considerable interest. In this paper we address ourselves to the topics of analysis and optimization of such nets. Those we consider are of the store-and-forward type more commonly known as message-switching networks.

The problem confronting the network designer is to create a system which provides suitable network performance at an acceptable system cost. Since, in message-switched networks the messages experience queueing delays as they pass from node to node, the performance measure is usually taken to be the speed at which messages can be delivered. The optimization problem is to achieve minimal average delay at a fixed network cost by appropriately choosing the network topology, the channel capacity assignment, and the message routing procedure. The purpose of this paper is to review some of the methods for handling various aspects of this problem.

II. ANALYTICAL TOOLS

The appropriate tools are those which have developed from queueing theory.

II.1. Single Server Systems. Much of queueing theory considers systems in which messages (customers) place demands for transmission upon a single communication channel (the single server). When the average demand for service is less than the capacity of the channel to handle these demands, the system is said to be stable. The literature on stable single server queueing systems is fairly voluminous as for example exemplified by the excellent work by Cohen [1]. Single server systems are characterized by A(t), the distribution of interarrival times and B(x), the distribution of service times. In the case when A(t) is exponential (i.e. Poisson arrivals), then the literature contains fairly complete results. However, when both A(t) and B(x) are arbitrary, then the situation becomes much more complex and only weak results are available.

Recently attention has been directed to developing approximate solution methods. These methods include: placing bounds on the behavior of the system; studying the system behavior under light and heavy traffic conditions; and by forming diffusion approximations to the physical queueing systems. This last approach appears most promising and involves replacing a discrete random process with a continuous random walk typically with a reflecting barrier at the origin to prevent queue sizes and waiting times from going negative (see Gaver [2]). Numerical results which have been obtained using the diffusion approximation have been startling in terms of their accuracy when compared to the original queueing problem.

II.2. Multiple Nodes and Networks. The case of interest to this paper is that of multiple nodes in a network environment. The queueing problems encountered in networks are far more difficult than single server problems. The difficulty arises due to new phenomena which occur in networks, the most important of which is that traffic entering a node in the network is dependent upon traffic elsewhere in the network and on other nodes through which this traffic has passed. This difficulty manifests itself in that A(t) for a network node is no longer exponential. A second difficulty is the phenomenon of blocking which occurs when the finite storage capacity of a node becomes filled and the further reception of messages is temporarily prohibited. This then places a burden on neighboring nodes and they too tend to get blocked causing the effect to propagate in the network. This effect is probably one of the least understood queueing effects in the study of nets and has significant impact upon performance.

The problem in which customers are permitted to move among a collection of queueing stations in some random fashion was studied by Jackson [3]. His major result was to show, when the system is stable, that each node in the system could indeed be analyzed as a single queueing facility (under Markovian assumptions). This represents perhaps one of the first successful attempts at decomposing a network problem into a series of simpler single node problems. Another fundamental result which permits decomposition of queueing networks is due to Burke [4]; he showed that if A(t) and B(x) are exponential, then the departure times are also exponentially distributed. Thus, we preserve the Poisson nature of the traffic flow between network nodes.

The results referred to in the previous paragraph do not carry over trivially into message oriented communication or computer nets since messages maintain their lengths as they pass through the net. The first comprehensive treatment of communication nets was carried out by Kleinrock [5]. Fortunately, it could be shown for a wide variety of communication nets that it was possible to introduce an assumption which once again permitted a decomposition of the network into a collection of single nodes.

Using the simple structure of the linear equations of motion governing Markovian queues, Wallace [6] has developed a procedure for solving the system of equations numerically.

III. OPTIMIZATION TOOLS

Perhaps the first communications network optimization problem was posed and solved by Kleinrock [5] in which he assumed that the network topology and the channel traffic were known quantities. Also, he assumed that the traffic was Markovian (Poisson arrivals and exponential message lengths) and justified certain decomposition assumptions. For each channel the optimal assignment of capacity C_i was found which minimized the average network delay T to messages, at a fixed total system cost D. We define: T_i as the average queueing plus transmission time on

*This work was supported by the Advanced Research Projects Agency, Dept. of Defense #DAHC15-69-C-0285.

the i^{th} channel; λ_i as the average message traffic on the i^{th} channel; γ as the average network traffic throughput; and d_i as the cost factor on the i^{th} channel. This problem takes the following form:

Problem A: Choose the set of channel capacities, C_i, to minimize T at fixed cost D where

$$T = \sum_i (\lambda_i/\gamma) T_i \quad (1); \quad D = \sum_i d_i C_i \quad (2)$$

The solution to this problem assigns a capacity to the i^{th} channel in an amount equal to the average traffic carried plus an excess capacity proportional to the square root of that traffic. It may be observed that a related capacity assignment (namely, that which gives capacity directly in proportion to traffic carried) provides an average message delay not significantly worse than the optimum.

These, and other related results, were published as Kleinrock's Ph.D. thesis (MIT) in 1962 (this work later appeared as [5]). Little was published in this field from then until 1969 [7]. Whatever the reason for this inactivity, it is clear that the recent interest is due to the development of computer networks. In 1967 Roberts [8] proposed the idea of an experimental computer network which later developed into the Advanced Research Projects Agency (ARPA) computer network (recently reported upon in the 1970 SJCC Proceedings).

In a forthcoming paper by Meister et al. [9], it is observed that in minimizing T in Problem A above, certain of the channels produce rather large and undesirable message delays T_i. As a result, Meister et al. pose the following problem, whose solution is closely related to that of Problem A:

Problem B: Same as Problem A except T is given by

$$T = [\sum_i (\lambda_i/\gamma) T_i^k]^{1/k} \quad (3)$$

By raising T_i to the k^{th} power they find that for $k > 1$, one forces a reduction in the variation among the T_i. For $k \to \infty$ the minimization yields a constant value for T_i. When $k = 0$, the assignment reduces to the proportional channel capacity assignment. The amazing observation is that T increases very slowly as k grows from unity. Moreover, they show that the variance of message delay is minimized when k is chosen equal to 2.

In Ref. [7] Kleinrock introduced some first attempts at modelling computer nets and was able to show that simple models were extremely useful in predicting the behavior of the message delay in the ARPA computer net. In a subsequent paper [10] he introduced the following variation to Problem A since the cost function as given in Eq. (2) was found not to represent tariffs for high speed telephone data channels:

Problem C: Same as Problem A except D is given by

$$D = \sum d_i C_i^{\alpha} \quad (4)$$

where $0 \le \alpha \le 1$. The solution to Problem C cannot be given in closed form. Nevertheless, in applying a numerical solution of this problem to the ARPA net it was found that the message delay varied insignificantly with α for $.3 \le \alpha \le 1$. This indicates that the closed form solution to Problem A may serve as an approximation to the more difficult Problem C.

IV. ADDITIONAL CONSIDERATIONS

Minimizing cost at fixed average message delay by appropriately choosing channel capacity is the dual for problems A, B, and C. This was studied in [10] and considered recently by Whitney [11]. Choice of network topology was considered in Kleinrock's original work. Recently Frank et al. [12] considered this problem for the ARPA net and developed suboptimal search procedures. They also addressed the problem of choosing an optimal channel assignment when capacities must be chosen from a finite set; Whitney [11] and Doll [13] considered this problem for a fixed tree topology. Frank et al. [14] devised an optimal procedure for selecting discrete channel capacities for centralized computer networks.

Message routing procedures must also be considered. Of all those so far discussed, this problem lends itself least to analysis. Lastly, we note that the ultimate standard in these problems is measurement of real systems. This is receiving considerable attention in the ARPA net.

V. CONCLUSION

The attempt in this paper has been to describe and to evaluate various tools for studying delay in communication and computer nets. These tools must be considerably improved. Nevertheless, they have been useful in network studies. Among the most difficult remaining problems we mention the blocking effect due to finite storage capacity, the analyses of routing procedures, and the design of network topologies.

REFERENCES

1. J. Cohen, The Single Server Queue, Wiley (1969).
2. D. Gaver, "Diffusion Approximations and Models for Certain Congestion Problems," JAP, 5, 1968.
3. J. Jackson, "Networks of Waiting Lines," Oper. Res., 5, 1957.
4. P. Burke, "The Output of a Queueing System," Oper. Res., 4, 1956.
5. L. Kleinrock, Communication Nets, McGraw-Hill (1964).
6. V. Wallace, "Representation of Markovian Systems by Network Models," SEL TR42, U. of Mich., 1969.
7. L. Kleinrock, "Models for Computer Networks," Proc. ICC, Boulder Colo., 1969.
8. L. Roberts, "Multiple Computer Networks and Intercomputer Communications," ACM Symposium on Operating Systems Principles, Gatlinburg, Tenn., 1967.
9. B. Meister, H. Mueller, and H. Rudin, "New Optimization Criteria for Message-Switching Networks," IBM Zurich Res. Lab., Switzerland, 1970.
10. L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," Proc. SJCC, May 1970.
11. V. Whitney, "A Study of Optimal File Assignment and Communication Network Configuration in Remote-Access Computer Message Processing and Communication Systems," SEL TR 48, U. of Mich., 1970.
12. H. Frank, I. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," Proc. SJCC, May 1970.
13. D. Doll, "Efficient Allocation of Resources in Centralized Computer-Communication Network Design," SEL TR 36, U. of Mich., 1969.
14. H. Frank, I. Frisch, W. Chou, and R. Van Slyke, "Optimal Design of Centralized Computer Networks," Proc. ICC, San Francisco, 1970.